

User's Manual  
DISK JOCKEY™ I CONTROLLER

Table of Contents

Introduction	1
Programming Specifications	3
Software Specifications	9
Disk System Software	14
I/O Connectors	16
Patches for CP/M™*	17
CP/M Cold Start Loader	21
CBIOS Drivers for CP/M	23
Hardware Level Registers	26
Parts List	30
Assembly Instructions	33
Power-Up & System Check Out	37
Hardware Description	44
Disk Jockey Firmware	56
Acknowledgment	63
Warranty	63
Schematic Drawings	

Copyright 1978 G. Morrow

**MORROW DESIGNS** 

User's Manual  
DISK JOCKEY™ I

INTRODUCTION

The Morrow's Micro-Stuff DISK JOCKEY I (DJ) board features two distinct subsections:

1. A universal floppy disk controller capable of interfacing to a wide variety of floppy disk drives,
2. A serial interface that allows communication with a terminal device at TTY 20ma current loop or RS-232 levels.

The floppy disk controller section will interface with any floppy disk drive plug compatible with the Shugart 800 drive. Siemens, Remex, Memorex and MFE are some of the manufacturers of full-sized floppy disks which are plug compatible with the Shugart 800/801.

The DJ plugs into an S-100 bus slot in a system with an 8080 type CPU and has a cable connector for a flat cable to the first floppy disk drive. The controller can control a chain of up to four drives daisy chained on this cable. A second connector on the DJ is provided for the attachment of a terminal device.

The DJ uses memory mapped I/O. Device registers used to input from and output to the floppy disk and the serial port are accessed from the CPU board of the S-100 system by references to memory addresses. Some registers differ in function depending on whether they are being read from or written to.

Most users will not wish to use the hardware level registers directly. Instead, they can call standard disk and serial I/O subroutines contained in 512 bytes of PROM memory on the DJ board. This PROM occupies a 512 byte block of S-100 bus memory addresses. A 256 byte RAM is also provided which is used by the PROM firmware as a data buffer and for temporary computation.

## Introduction

The actual addresses where the I/O registers, PROM, and RAM appear is controlled by another PROM, referred to as the address selection PROM. This PROM is supplied with standard addresses burned into it for these registers. If the standard addresses would conflict with some other device on the system, a PROM burned with non-standard addresses can be substituted.

The Disk Jockey uses 1024 bytes of memory starting at 340:000Q or E000H (standard version). The first 512 bytes are occupied by ROM, the next 256 bytes contain a RAM buffer. The last 256 bytes constitute the memory mapped I/O.

## PROGRAMMING SPECIFICATION

Most users will wish to take advantage of the standard I/O sub-routines supplied in PROM on the DJ.

The user should branch to the appropriate address in a jump table in the first few words of the system ROM. Since the subroutines end with a RET instruction, a CALL instruction should be used to branch to the subroutine.

The jump table contains jump instructions to the true address of the utility routines within the ROM. Having a jump table allows the individual routines to be updated and moved around within the ROM without having to change software that calls the routines. Let A represent the address of word 0 of the onboard ROM. In boards with standard address decoding PROMS, A = 340:000. The addresses to call for the utility routines are then:

Address	Standard Value	Symbolic Value	Function
A	340:000	DBOOT	DOS bootstrap routine
A+3	340:003	TERMIN	Serial input
A+6	340:006	TRMOUT	Serial output
A+9	340:011	TKZERO	Recalibrate (Seek to TRK00)
A+12	340:014	TRKSET	Seek
A+15	340:017	SECTOR	Select sector
A+18	340:022	DMA	Set DMA address
A+21	340:025	READ	Read a sector of disk data
A+24	340:030	WRITE	Write a sector of disk data

The specific function of each subroutine is described below.

A subroutine upon completion will execute a RET instruction. A disk subroutine that completes normally will return with a zero in the accumulator. A disk subroutine that detects an error condition will return a nonzero value in the accumulator. A program should execute an ANA A instruction after return from a disk utility subroutine to set the flags according to the contents of the accumulator and then branch if non-zero to an appropriate error handling routine.

### Serial I/O

At the hardware level, a means is provided for the CPU program to determine whether the present level of the serial input is a MARK (1) or a SPACE (0). The user has a choice of connecting to the TTY or RS-232 serial input pin depending on the electrical requirements of the input device.

When a program in the CPU looks at bit 1 of the Read Status Register (see Hardware Registers below), this bit will be a 0 if the input line is at the SPACE level and a 1 if the input line is at the MARK level.

## Programming Specification

The hardware only provides the facility for the software to determine whether the device attached to the serial in line is sending a 0 or 1 signal. Decoding a pattern in time of zeroes and ones into meaningful information is the province of software.

Built-in Disk Jockey firmware provides the facility to receive or send a byte of data from the input device according to industry wide conventions for serial data transmission. These conventions are explained below.

Normally, the interface line will be at a 1 level. When an input device wishes to send a byte of data, it puts the input line to a 0 level for a length of time referred to as one bit time. This is called the start bit. At the conclusion of the start bit, the input device sends in order starting with the least significant bit and ending with the most significant bit, the eight data bits of the byte. To do this, the input device puts the interface line at a 0 or a 1 level for one bit time, depending on the value of the data bit to be sent. At the conclusion of the bit time, the next data bit is sent. After the final bit is sent, the line returns to the 1 level for at least one bit time (the stop bit).

After the final stop bit is sent, the input device is free to send the next character. If the next character is not yet available, for example, the next key on the input keyboard has not been pressed, the input device can wait any amount of time before lowering the line to a 0 to signal the beginning of the next character.

The format described above is the most common format used by terminal devices. There is some variation, however. Some devices use two stop bits, some transmit a parity bit, etc. Check the documentation for the specific terminal in use for details.

Different input devices can send their information at different baud rates. The relationship between baud rate and bit times is given below. Many terminal devices have switch selectable baud rates. The user will typically wish to select the highest baud rate that will not exceed the capacity of the transmission path between the terminal and the Disk Jockey.

Baud Rate	Bit Times
110	9.091 ms
300	3.333 ms
1200	.833 ms
1800	.555 ms
2400	.417 ms
4800	.208 ms

## Programming Specification

A program assembling an incoming character will typically wait one-half bit time from the leading edge of the start bit and then check to see if the start bit is still present. It will then wait for one bit time and sample the first data bit, etc. This technique samples each bit in the middle of its bit time, greatly improving the noise immunity of the system.

The firmware supplied with the Disk Jockey will assemble a byte received from the interface using this technique. A timing loop generates the delay of one-half bit time or one bit time as required. The timing constant used by this loop must be set by the user according to the baud rate required. The value of the constant also depends on the instruction execution time of the processor.

To set the timing constant, a program must deposit the correct value appropriate to the baud rate and processor speed at a fixed location in the Disk Jockey's onboard RAM. The address of this location has the symbolic name SCON in the firmware listing. Its value in units with a standard decoding ROM is 342:160. The value of the timing constant that should be stored for commonly used baud rates is given in the following table for a 2 Mhz clock system:

Baud Rate	SCON decimal	SCON octal	SCON hex
110	375	1:166	176
300	135	207	87
1200	33	41	21
1800	21	25	15
2400	16	20	10
4800	7	7	7

This timing constant must be set before the serial input or serial output subroutine is called.

The subroutine TERMIN can be called to wait for and assemble a character arriving over the serial input line. The character, once it is assembled, will be returned in CPU register A. The subroutine will not return until a character arrives.

The subroutine TRMOUT will transmit out the serial output port the character in register A. It will return once the character has been transmitted.

It is important to be aware of some of the disadvantages of serial communications as they affect programming technique. Since the CPU must sample the input line at the correct time and be listening to detect the start bit, the serial input subroutine must be called before the

## Programming Specification

start bit is received and must stay in its timing loops until a character is assembled. Once a character is received, all computation done by the program that called for the character must be done and control returned to the serial input subroutine before the start of the next character. This is normally not a problem due to the CPU instruction times being so short with respect to the bit times for commonly used baud rates, but attention must sometimes be paid to this requirement. Another disadvantage of this technique is the requirement to tie up the CPU to listen to see if a character is arriving. This prohibits a program that runs continuously until a key is pressed unless the program can guarantee that it can check for a start bit sufficiently frequently while doing the other computation that is necessary.

The performance of the serial input system can be improved when it becomes desirable to do so either by adding a more complex serial interface including a UART chip that will assemble an entire character without requiring intervention from the CPU or by using a parallel interface with a flag bit.

In spite of the disadvantages listed, serial communications offers the most economical way to attach a terminal to a computer. Virtually every terminal device manufactured offers a serial data path for connection to its CPU.

### Disk I/O

To understand the significance of the disk utility subroutines, it is necessary to say a few words about how data is organized on the disk.

Information on the disk is organized into a number of concentric tracks:

Full-sized floppy	77 tracks
Mini-floppy	35 tracks

The disk read/write head can be moved to any track by a series of step in or step out commands. A step in command moves the read/write head one track towards the center of the disk. A step out command moves the head one track away from the center of the disk. It is the responsibility of software to keep track of what track number the disk is currently at and to calculate how many step in or step out commands are necessary to move the head to a desired new position.

Once the read/write head has been moved to the desired track, the rotation of the disk will move a circle of magnetic material beneath the head. Within this circle of material, data is recorded in distinct regions called sectors. The sector is the smallest amount of information that can be separately read or written from the disk. There are twenty-six sectors/track on a standard and sixteen sectors/track on a mini-floppy. Each sector contains 128 data bytes.

## Programming Specification

In the header field of each sector, the track and sector number is recorded. During read or write commands, this header is read before data transfer takes place. If the DJ firmware detects a discrepancy between the track number it thinks it is at and the number recorded on the disk, it reports an error. To recover from this error, a program must do a recalibrate operation. The disk drive has a sensor that reports when the disk is physically positioned at track 00. A series of step out commands must be issued until this status line comes on. This operation will always position the disk to the same physical track. The recalibrate sequence is a standard utility subroutine supplied with the disk firmware.

Transferring a sector of disk data between memory and the disk therefore involves the following steps, each corresponding to a subroutine call to Disk Jockey firmware:

Position the read/write head to the desired track.

Specify the sector number to be involved in the data transfer.

Specify the memory address that disk data is to be written from or read to.

Actually perform the read or write operation.

Check for error conditions.

### Subroutines

- TRKSET - Given a one byte track address between 0 and 76 in register C of the CPU, this subroutine verifies that it is a valid track address. It then performs a seek operation to position the read/write head to the desired track. It does not read the header fields of the track to verify that the seek took place correctly -- this is done when an attempt is made to read or write a sector.
- SECTOR - Given a one byte sector number between 1 and 26 in register C, this subroutine checks that it is a valid sector number. It then records it for use in a later read or write.
- DMA - Given a memory address in the B-C register pair, this subroutine checks that it is not an address within the Disk Jockey's address space. It then records the address in its RAM. Any subsequent read or write operations will transfer data to or from this memory address (and the 127 successively higher numbered memory cells). No disk operations actually take place.
- READ - 128 bytes of data are read off the current track from the sector specified by the last SECTOR subroutine call to the address in memory specified by the last DMA call.
- WRITE - 128 bytes of data are written on the current track into the sector specified by the last sector subroutine call from the address in memory specified by the last DMA call.



## PROGRAMMING SPECIFICATION

- TKZERO - Calling this subroutine will position the read/write head to track 00 using the track 00 sensor as a reference. As discussed above, a call to this subroutine is the appropriate response to a seek error -- that is, the software detecting that the track number recorded in the header field of a sector does not agree with the count of track number being maintained in software.
- DBOOT - Branching to this routine will result in a bootstrap load operation from the floppy disk. The serial constant is initialized to 1200 baud. Sector 1 of track 00 of disk 1 will be read into memory location 200Q (80H). A branch will then be performed to this location. Branching to this subroutine from the front panel is the typical way to initialize a disk system.
- DISKETTE INITIALIZATION - Before a new diskette can be successfully used, it must be initialized. Most diskettes are sold pre-initialized. However, it is sometimes necessary to reinitialize a diskette. The initialization process involves writing the header field of every sector onto the disk. None of the subroutines described above can be used to write these header fields. This is a safety measure to ensure that an erroneous branch to the firmware PROM cannot reinitialize a disk, destroying all the data recorded on it. The initialization function for diskettes is typically provided by a command included in the Disk Operating System. CP/M diskettes furnished by Thinker Toys contains a command INTLIZE.COM. The source code for this command, INTLIZE.ASM is also included. The Disk/ATE diskette has a command file called DSKINT which initializes diskettes and places a bootstrap loader on track zero sector one.

## SOFTWARE SPECIFICATIONS

### The Bootstrap

At location 340:000Q (E000H), there is a bootstrap routine which is intended to make the disk come up automatically. This routine selects drive number 1, does a home to track 00 and reads sector 1 of track 00 into locations 200Q through 377Q. After reading this sector into memory, the routine branches to location 200Q. The first sector of track 00 should have a loader which will bring in the rest of the user's system. Customers who purchase CP/M with The Disk Jockey receive a diskette which has this loader. By setting the program counter of the CPU to 340:000Q (E000H) and pressing RUN, CP/M automatically loads itself into memory and starts.

### Full-Sized Floppy/Mini-Floppy Hardware Configuration

The Disk Jockey controller has two slide switches which configure the board hardware to control either an 8" drive or a 5" drive.

8" Full-Sized Floppy Drives: BOTH slide switches must be in the LEFT-most position;

5" Mini-Floppy Drives: BOTH slide switches must be in the RIGHT-most position.

### Selecting Drives

If decoders are installed in the disk drives themselves, up to eight drives can be connected in a daisy chain fashion to the Disk Jockey. Without decoders, four drives can be controlled by the Disk Jockey. It is important to note that none of the firmware with the exception of the boot loader ever changes the selection of drives. Before sending commands via the firmware, it is the programmer's responsibility to select the proper drive by storing the drive selection code in two places in memory.

Drive selection codes for drives 1, 2, 3 and 4 are as follows:

Drive	Drive Selection Code	
1	210Q	88H
2	011Q	09H
3	050Q	28H
4	012Q	0AH

The appropriate drive selection code must be stored at memory locations 342:163Q (E273H) and 343:002 (E302H).

## Soft Ware Specifications

### Utilizing the Disk Jockey Firmware

In order to transfer information to and from a disk drive under the control of the Disk Jockey, the firmware functions must be used in the proper sequence. A seek function should always precede a set sector function. A read or write function should be preceded by a DMA function, a set sector function and a seek function.

### Data Transfer Examples

READ:

Suppose sectors 5, 6, 7 and 8 of track 12, drive 1, need to be read into memory starting at location 7:200Q (780H). The following program will do this:

Octal

100	076	210	READ	MVI	A,210Q	INITIALIZE
102	062	163		STA	DRIVE	DRIVE
105	062	002		STA	FUNCTN	#1
110	061	150		LXI	SP,STACK	INITIALIZE STACK POINTER
113	315	011		CALL	TKZERO	RECALIBRATE HEAD
116	016	014		MVI	C,12	SEEK HEAD TO
120	315	014		CALL	TSEEK	TRACK 12
123	001	005		LXI	B,4:005Q	SECTOR COUNT & NUMBER
126	041	200		LXI	H,7:200Q	DMA ADDRESS
131	305		LOOP	PUSH	B	SAVE SECTOR & COUNT
132	345			PUSH	H	SAVE DMA ADDRESS
133	315	017		CALL	SECTOR	SET THE SECTOR
136	301			POP	B	GET DMA ADDRESS
137	305			PUSH	B	SAVE IT AGAIN
140	315	022		CALL	DMA	SET DMA ADDRESS
143	315	025		CALL	DISKR	READ DISK SECTOR
146	247			ANA	A	TEST FOR ERRORS
147	302	100		JNZ	READ	START OVER IF ERROR
152	341			POP	H	RECOVER DMA ADDRESS
153	301			POP	B	RECOVER SECTOR & COUNT
154	005			DCR	B	DECREMENT COUNT
155	310			RZ		RETURN IF DONE
156	014			INR	C	INCREMENT SECTOR
157	021	200		LXI	D,200Q	INCREMENT DMA
162	031			DAD	D	ADDRESS BY 200Q
163	303	131		JMP	LOOP	READ NEXT SECTOR

Hex

40	3E	88	START	MVI	A,88H
42	32	73		STA	DRIVE
45	32	02		STA	FUNCTN
48	31	68		LXI	SP,STACK
4B	CD	09		CALL	TKZERO

## Soft Ware Specifications

4E	0E 0C		MVI	C,12
50	CD 0C E0		CALL	TSEEK
53	01 05 04		LXI	B,405H
56	21 80 07		LXI	H,780H
59	C5	LOOP	PUSH	B
5A	E5		PUSH	H
5B	CD 0F E0		CALL	SECTOR
5E	C1		POP	B
5F	C5		PUSH	B
60	CD 12 E0		CALL	DMA
63	CD 15 E0		CALL	DISKR
66	A7		ANA	A
67	C2 40 00		JNZ	START
6A	E1		POP	H
6B	C1		POP	B
6C	05		DCR	B
6D	C8		RZ	
6E	0C		INR	C
6F	11 80 00		LXI	D,80H
72	19		DAD	D
73	C3 59		JMP	LOOP

### WRITE:

The following program writes from memory starting at location 200:000Q (8000H) onto tracks 4, 5, and 6 of disk drive 1.

### Octal

200	076 210	WRITE	MVI	A,210Q	INITIALIZE
202	062 163 342		STA	DRIVE	DRIVE
205	062 002 343		STA	FUNCTN	#1
210	061 150 342		LXI	SP,STACK	INITIALIZE STACK POINTER
213	315 011 340		CALL	TKZERO	RECALIBRATE DISK HEAD
216	076 004		MVI	A,4	SET UP
220	062 305 000	TLOOP	STA	TEMP	TRACK REGISTER
223	117		MOV	C,A	SEEK TO
224	315 014 340		CALL	TSEEK	NEXT TRACK
227	001 001 032		LXI	B,32:001Q	SECTOR COUNT & NUMBER
232	041 000 200		LXI	H,200:000	DMA ADDRESS
235	305	SLOOP	PUSH	B	SAVE SECTOR & COUNT
236	345		PUSH	H	SAVE DMA ADDRESS
237	315 017 340		CALL	SECTOR	SET CURRENT SECTOR
242	301		POP	B	RECOVER
243	305		PUSH	B	DMA ADDRESS & SAVE
246	315 022 340		CALL	DMA	SET DMA ADDRESS
251	315 030 340		CALL	DISKW	WRITE A SECTOR
254	247		ANA	A	TEST FOR WRITE
255	302 255 000	STALL	JNZ	STALL	PROTECTED DISKETTE
260	341		POP	H	RECOVER DMA ADDRESS
261	301		POP	B	RECOVER SECTOR & COUNT
262	021 200 000		LXI	D,200Q	INCREMENT
265	031		DAD	D	DMA ADDRESS

# Soft Ware Specifications

266	014		INR	C	INCREMENT SECTOR
267	005		DCR	B	DECREMENT COUNT
270	302 235 000		JNZ	SLOOP	READ ANOTHER SECTOR
273	072 305 000		LDA	TEMP	GET OLD TRACK
276	074		INR	A	& INCREMENT
277	376 007		CPI	7	TEST FOR
301	310		RZ		DONE
302	303 220 000		JMP	TLOOP	SEEK TO NEXT TRACK
305	000	TEMP:	DB	0	TEMPORARY TRACK REG

## Hex

80	3E 88	WRITE	MVI	A,88H
82	32 73 E2		STA	DRIVE
85	32 02 E3		STA	FUNCTN
88	31 68 E2		LXI	SP,STACK
8B	CD 09 E0		CALL	TKZERO
8E	3E 04		MVI	A,4
90	32 C5 00	TLOOP	STA	TEMP
93	4F		MOV	C,A
94	CD 0C E0		CALL	TSEEK
97	01 01 1A		LXI	B,1A01H
9A	21 00 80		LXI	H,8000H
9D	C5	SLOOP	PUSH	B
9E	E5		PUSH	H
9F	CD 0F 34		CALL	SECTOR
A2	C1		POP	B
A3	C5		PUSH	B
A6	CD 12 34		CALL	DMA
A9	CD 18 34		CALL	DISKW
AC	A7		ANA	
AD	C2 AD 00	STALL	JNZ	STALL
B0	E1		POP	H
B1	C1		POP	B
B2	11 80 00		LXI	B,80H
B5	19		DAD	D
B6	0C		INR	C
B7	05		DCR	B
B8	C2 9D 00		JNZ	SLOOP
BB	3A C5 00		LDA	TEMP
BE	3C		INR	A
BF	FE 07		CPI	7
C1	C8		RZ	
C2	C3 90 00		JMP	TLOOP
C5	00	TEMP:	DB	0

## Soft Ware Specifications

The two examples above will appear to run rather slowly. This is due to reads and writes being done on consecutive sectors. Since there is overhead error checking done in the disk read and write firmware, the next sector has been passed over by the time the subsequent read or write operation commences. Therefore, the disk has to go almost a full revolution before the right sector is under the head again. Thus, to write twenty-six sectors, the diskette revolves roughly twenty-six revolutions instead of one. The method used to overcome this problem is called reading skewed sectors. That is, instead of reading, say, sectors 1, 2, 3, 4, 5 ..., 26, in consecutive order, it is more efficient to read in the order 1, 6, 11, 16, 21, 26, 5, 10, 15, 20, 25, 4, 9, 14, 19, 24, 3, 8, 13, 18, 23, 2, 7, 12, 17, 22. This will read an entire track in five revolutions instead of twenty-six. For an example of how to read and write sectors with a skew of five, see the software listings for patching CP/M\* to the Disk Jockey.

\*CP/M is a trade mark of Digital Research, Pacific Grove, CA.

## DISK SYSTEM SOFTWARE

An assembled Disk Jockey is part of a DISCUS I system and is also accompanied by a copy of Disk/ATE<sup>tm</sup>. Disk/ATE and CP/M\* are also available at additional cost to those who have purchased only the controller. Both Disk/ATE and Disk Jockey CP/M are tailored to the I/O of the Disk Jockey controller. Both expect that a serial TTY/RS232 terminal set for 1200 baud is connected to J2 (serial port) of the Disk Jockey. Both are supplied on a write protected diskette (notch open) which should be kept that way. DO NOT COVER THE NOTCH ON THE DISKETTE. Finally, both systems are designed to self load when the disk is in place in drive A and a branch to 340:000Q (E000H) is made. For the CP/M user, a series of manuals accompanies the diskette which describes how to back-up the CP/M diskette. The only precaution is that when drive B is to be used for back-up purposes, it must be "logged on" (e.g., DIR:B) before the back-up process is started.

### Backing Up Disk/ATE

To make a back-up copy of Disk/ATE, load Disk/ATE and have a blank diskette which is not write protected (notch covered). Follow the steps outlined below:

- a. Type: B16  
followed by a carriage return. This command forces ATE to express numbers and addresses in hexadecimal radix.
- b. Type: L IOTBL <T>  
followed by a carriage return. This command loads the I/O driver symbol table from the disk.
- c. Type: ? SYSIO...IOEND  
followed by a carriage return. This is the beginning and ending addresses of the ATE I/O driver software expressed in hexadecimal radix. Make a note of these two values.
- d. Type: L ATETBL <T>  
followed by a carriage return. This command loads a selected subset of ATE's symbol table from the disk.
- e. Type: ? BEGIN...END  
followed by a carriage return. This is the beginning and ending address of ATE expressed in hexadecimal radix. Make a note of these two values.
- f. Type: GO DSKINT  
followed by a carriage return. This is the load and go command for the diskette initialization program that formats a diskette to the IBM soft-sectored standard and also places a boot strap loader on track zero sector one.

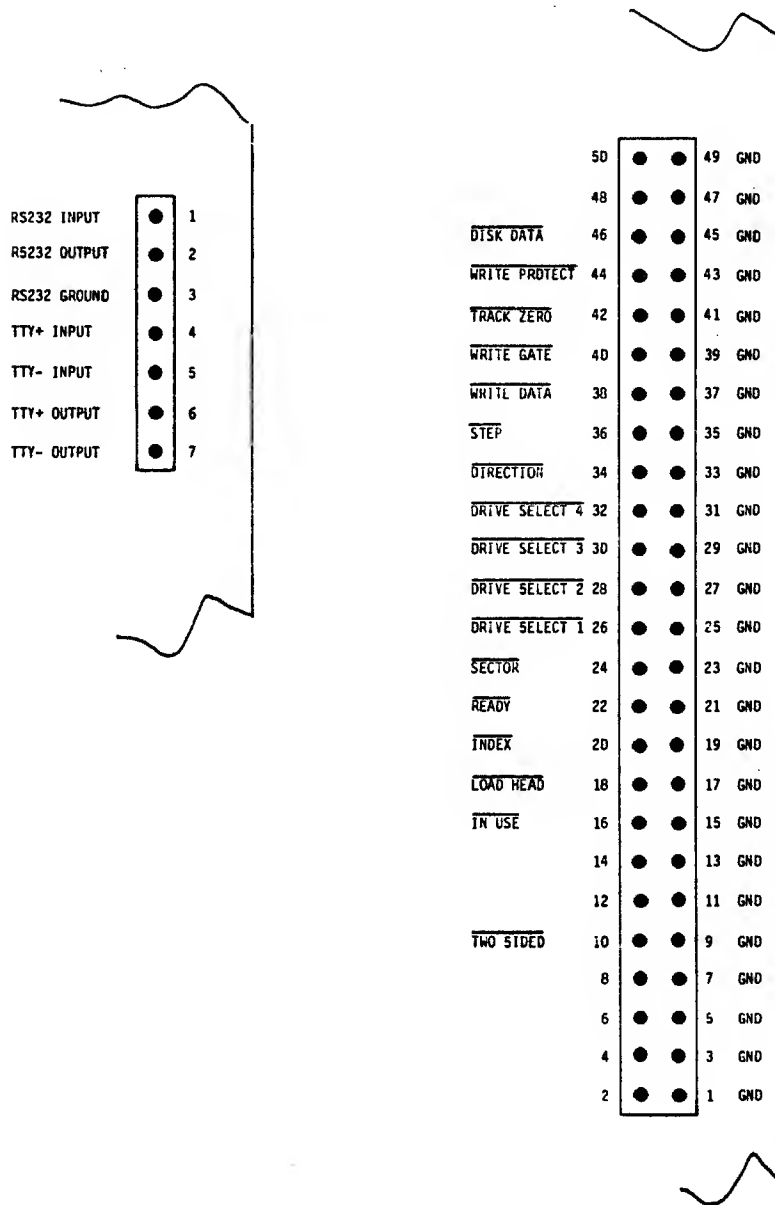
## Disk System Software

- g. DSKINT will ask that the diskette be placed in the drive and then will ask which drive the diskette is in. If the diskette is write protected or the diskette is not in the drive or if the drive door is not closed or if the indicated drive is not connected to the system, DSKINT will detect these conditions and start all over until all the conditions necessary to write on a diskette are satisfied. After completing the initialization, DSKINT automatically returns to ATE.
- h. IO and ATE must be saved on the new diskette. IO must be saved first.
- i. Using the values for SYSIO and IOEND obtained in step c, type:  
S IO (SYSIO value here)H...(IO END value here)H  
followed by a carriage return. The "H" suffix is necessary to force ATE to interpret the preceding digits as a number in hexadecimal radix.
- j. Using the values for BEGIN and END obtained in step e, type:  
S ATE (BEGIN value here)H...(END value here)H  
followed by a carriage return.
- k. Disk/ATE has now been copied on the fresh diskette. Files may now be transferred from the original diskette as needed.



## I/O CONNECTORS J1 AND J2

Illustrated below are the details of the pin connections of J1 and J2. In both illustrations, the top of the circuit board is shown as the straight line on the right side of the connector.



## PATCHES FOR CP/M\*

### General

This section is included for those users of the Disk Jockey who have purchased a copy of CP/M Vers. 1.4 from a source other than Thinker Toys. Copies of CP/M sold through Thinker Toys have the necessary I/O routines to interface CP/M to the Disk Jockey disk controller and serial I/O port.

At the end of this section are two listings which are designed to allow the Disk Jockey to be interfaced with the Digital Research CP/M operating system. This can be done with a minimum of effort.

The first listing is the so called "cold start loader" which is used to bring CP/M in from the disk. It also has code which will allow the user to easily write a modified version of CP/M out on the disk. There is even a small routine which loads the "cold start loader" itself on sector 1 of track 0.

The second listing is CBIOS software (Custom Basic Input-Output System) which is the interface between CP/M and the Disk Jockey controller. The general idea is to key in the cold start loader, use the loader to bring CP/M in from a diskette, enter the CBIOS code and, finally, use the cold start loader to save everything out on a clean diskette.

### The "Cold Start Loader"

There are three parts to the cold start loader. LOAD is at address 200Q (80H) and is designed to read CP/M into memory from location 51:000 through 77:377 (290H through 3FFFH). After loading CP/M, the LOAD routine branches to location 76:000 which is a routine that initializes several memory locations, prints a sign-on message and then branches to CP/M proper.

SAVE is at location 232Q (9AH) and is the reverse of LOAD. SAVE writes out on the disk starting at track 0 sector 2 all memory locations between 51:000 and 77:377. After performing this operation, SAVE comes to a dynamic halt at STALL 261 (B1H).

INTLZ is a short routine which writes locations 200 through 377 on sector 1 of track 0. Thus, once the cold start loader is keyed into memory, it can save itself at the right location on the disk.

### CBIOS

The standard version of CP/M is designed to run with the Intel MDS development system a floppy disk interface. Most of the CP/M system software is completely independent of the particular 8080 hardware environment that it is running in. However, there is a certain part which must be tailored to the hardware of the host system. This hardware-dependent software is completely contained on pages 76 and 77 of CP/M (assuming the standard 16K version). CP/M can be made to run on different hardware by changing the software on pages 76 and 77 (3E00H through 3FFFH). The CBIOS software which is supplied with the Disk Jockey is designed to let CP/M run when an 8" full-sized floppy disk is attached to the Disk Jockey controller which is plugged into an S-100 main frame.

## Patches for CP/M\*

### Patching CP/M\*

Before actually performing any of the steps below, the Disk Jockey should be plugged into the mainframe S-100 bus, and an 8" disk drive should be connected to the controller. You should have on hand two diskettes: one with CP/M and one blank. A copy of CP/M which will run on the Disk Jockey will be constructed on the blank disk before any changes are attempted on the original CP/M disk.

#### Step I:

Plug in the controller. Connect the disk to the controller and turn on the CPU and disk drive. Do NOT put a diskette in the drive at this point.

#### Step II:

Enter the "cold start loader" into main memory starting at location 200 (80 H). The instructions will extend from 200 to 377 filling all of the second half of page 0.

#### Step III:

Load location 342:163 with 210 (E273 hex with 88 hex).

Load location 343:002 with 210 (E302 with 88 hex). Location 343:002 is in actuality an I/O device (memory mapped I/O) so values read from 343:002 and the values written into 343:002 will be independent.

#### Step IV:

Set the program counter of the CPU to location 272 (BA hex) but do NOT start the CPU yet.

#### Step V:

Insert the blank diskette into the drive and close the door. Be sure that the diskette is NOT write protected. (Write protected 8" diskettes have a notch near the corner of the diskette diagonally opposite the labeled corner.) Be sure the diskette is inserted right side up. On a Discus I system, the label should be on top and the diskette should be inserted by holding the label of the diskette between the thumb and forefinger when it is inserted into the drive.

## Patches for CP/M\*

### Step VI:

Start the computer. After the CPU is started at 272, the activity light (if one is present) should come on, the head should load and step out to track 0 (if not already there). After sixteen revolutions of the diskette, the head will unload and the activity light will go off.

### Step VII:

Stop the CPU. It should be executing the instruction JMP DONE. -- 303 312 000 (C3 CA 00 hex). The cold start loader should be on sector 1 of track 0.

### Step VIII:

Remove the diskette from the disk drive.

### Step IX:

Change locations 222 and 223 from 000 and 076 to 261 and 000 respectively (change 92 to B1 and 93 to 00 hex).

### Step X:

Initialize the program counter of the CPU to 200 (80 H). Do NOT start the CPU.

### Step XI:

Insert the CP/M disk and be sure that it is write protected before it is put into the drive. Close the drive door securely.

### Step XII:

Start the CPU. The head should load and after a second or two the head should step to track 1. Wait for the head to unload and the activity light to go off. CP/M has been loaded into memory between 51:000Q and 77:377Q.

### Step XIII:

Enter the CBIOS code starting at location 76:000 through 77:072 (3E00 through 3F3A hex). Verify that the code has been entered correctly.

## Patches for CP/M\*

### Step XIV:

Initialize the program counter of the CPU to 232 (9A hex) but do NOT start the CPU.

### Step XV:

Insert the blank (except for the "cold start loader") diskette into the drive and close the door securely.

### Step XVI:

Start the CPU. The head should load, return to track 0 and write the better part of tracks 0 and 1 before the head unloads. After the head unloads, do NOT remove the diskette but stop the CPU. The CPU should be executing the JMP STALL instruction.

### Step XVII:

Connect a terminal to the serial port of the Disk Jockey and adjust the baud rate to 1200 baud.

### Step XVIII:

Initialize the program counter of the CPU to location 340:000 (E000 hex) and start the CPU (the blank disk should still be in the drive). After several seconds, the terminal should print:

```
16k CP/M VERS/1.4
```

After five seconds or so, the prompt should appear:

```
A>
```

A Disk Jockey version of CP/M is now up and running. After this new version of CP/M has been tested (see CP/M documentation), Steps I through XVII can be used to alter the original CP/M diskette if desired.

"Cold Start Loader"

CP/M\* "COLD START LOADER"

```

342:156  STACK EQU 342:156Q
342:164  DMAADDR EQU 342:164Q
340:017  SECTOR EQU 340:017Q
340:025  READ EQU 340:025Q
340:030  WRITE EQU 340:030Q
340:014  SEEK EQU 340:014Q
340:011  HOME EQU 340:011Q
340:000  BOOTSTRAP EQU 340:000Q
    
```

```

000:200 315 315 000  LOAD  CALL  SETUP  INITIALIZE SECTOR COUNT
203 000          RLOOP  NOP      & DMA ADDRESS
204 305          PUSH  B      SAVE SECTOR & COUNT
205 315 017 340  CALL  SECTOR  INITIALIZE SECTOR
210 315 025 340  CALL  READ   READ DATA
213 247          ANA   A      TEST FOR ERROR
214 302 000 340  JNZ  BOOTSTRAP RESTART IF ERROR
217 301          POP  B      RECOVER SECTOR & COUNT
220 005          DCR  B      DECREMENT &
221 312 000 076  JZ   76:000Q    TEST FOR CP/M LOADED
224 315 327 000  CALL  NSDMA  CALCULATE NEXT SECTOR
227 303 203 000  JMP   RLOOP  & DMA ADDR
    
```

```

232 061 156 342  SAVE  LXI  SP,STACK  INITIALIZE STACK POINTER
235 315 011 340  CALL  HOME  RETURN HEAD TO TRACK 0
240 315 315 000  CALL  SETUP  INITIALIZE SECTOR, COUNT
243 000          WLOOP  NOP      & DMA ADDR
244 305          PUSH  B      SAVE SECTOR & COUNT
245 315 017 340  CALL  SECTOR  GET SECTOR ADDRESS
250 315 030 340  CALL  WRITE  WRITE DATA
253 247          ANA   A      TEST FOR
254 302 261 000  JNZ  STALL  ERROR
257 301          POP  B      RECOVER SECTOR & COUNT
260 005          OCR  B      DECREMENT COUNT
261 312 261 000  STALL JZ   STALL  DYNAMIC HALT
264 315 327 000  CALL  NSOMA  CALCULATE NEXT SECTOR
267 303 243 000  JMP   WLOOP  & DMA ADDR
    
```

```

272 061 156 342  INTLZ LXI  SP,STACK  INITIALIZE STACK POINTER
275 315 011 340  CALL  HOME  RETURN HEAD TO TRACK 0
300 041 200 000  LXI  H,200Q  INITIALIZE
303 042 164 342  SHLD DMAADDR  DMA ADDRESS
306 315 030 340  CALL  WRITE  WRITE 200Q THROUGH
311 000          NOP      377Q ON TRACK 0 SECTR 1
312 303 312 000  DONE  JMP   DONE  DYNAMIC HALT
    
```

```

000:315 001 006 056  SETUP LXI  8,56:006Q  SECTOR COUNT & INITIAL SECTR
320 041 000 053  LXI  H,53:000Q  INITIALIZE DMA
323 042 164 342  SHLD DMAADDR  ADDRESS5
326 311          RET

327 076 005          NSDMA MVI  A,5      SECTOR SKEW
331 201          AOD  C      AOD TO CURRENT
332 117          MOV  C,A    SECTOR
333 336 033          SBI  27  TEST FOR
335 372 357 000    JM   OK      SECTOR OVERFLOW
340 074          INR  A      ADJUST
341 117          MOV  C,A    NEW SECTOR
342 021 200 365    LXI  D,365:200Q  DMA ADJUSTMENT
345 376 001          CPI  1      TEST FOR SECTOR 1
347 302 362 000    JNZ  OK+3  NOT THROUGH W/ TRACK 0
352 305          PUSH B      THROUGH W/ TRACK 0
353 315 014 340  CALL  5EEK  MOVE TO TRACK 1
356 301          POP  B      RECOVER SECTOR & COUNT
357 021 200 002    LXI  D,2:200Q  DMA ADJUSTMENT
362 052 164 342  LHLD DMAADDR  GET OLD DMA ADDR
365 031          DAD  D      CALCULATE NEW
366 042 164 342  SHLD DMAADDR  DMA ADDR & SAVE
371 076 077          MVI  A,77Q  TEST FOR
373 224          SUB  H      ADDRESS OVERFLOW
374 372 327 000  JM   NSDMA  & ADJUST IF
377 311          RET      NECESSARY
    
```

"Cold Start Loader"

Hex Listings  
(see octal listings for comments)

CP/M\* "COLD START LOADER"

```

E26E      STACK EQU 342:156Q
E274      DMAADDR EQU 342:164Q
E00F      SECTOR EQU 340:017Q
E015      READ EQU 340:025Q
E018      WRITE EQU 340:030Q
E00C      SEEK EQU 340:014Q
E009      HOME EQU 340:011Q
E000      BOOTSTRAP EQU 340:000Q
    
```

```

0080      CD C0 00      LOAD      CALL      SETUP
0083      00              LOOP      NOP
0084      C5              PUSH      8
0085      C0 0F E0      CALL      SECTOR
0088      C0 15 E0      CALL      READ
0088      A7              ANA      A
008C      C2 00 E0      JNZ      BOOTSTRAP
008F      C1              POP      8
0090      05              DCR      8
0091      CA 00 3E      JZ       76:000Q
0094      CD D7 00      CALL      NSDMA
0097      C3 83 00      JMP      RLOOP

009A      31 6E E2      SAVE     LXI      SP,STACK
009D      CD 09 E0      CALL     HOME
00A0      C0 CD 00      CALL     SETUP
00A3      00              WLOOP   NOP
00A4      C5              PUSH     8
00A5      CD 0F E0      CALL     SECTOR
00A8      CD 18 E0      CALL     WRITE
00AB      A7              ANA     A
00AC      C2 B1 00      JNZ     STALL
00AF      C1              POP     8
00B0      05              OCR     8
00B1      CA 81 00      STALL   JZ       STALL
00B4      CD D7 00      CALL     NSDMA
00B7      C3 A3 00      JMP     WLOOP

00BA      31 6E E2      INTLZ   LXI     SP,STACK
00B0      CD 09 E0      CALL     HOME
00C0      21 80 00      LXI     H,200Q
00C3      22 74 E2      SHLD    OMAADDR
00C6      CD 18 E0      CALL     WRITE
00C9      00              NOP
00CA      C3 CA 00      DONE   JMP     DONE
    
```

```

00CD      01 06 2E      SETUP   LXI     8,56:006Q
00D0      21 00 2B      LXI     H,53:000Q
00D3      22 74 E2      SHLD    DMAADDR
00D6      C9              RET

00D7      3E 05              NSDMA   MVI     A,5
00D9      81              ADD     C
00DA      4F              MOV     C,A
00DB      DE 1B              SBI     27
00DD      FA EF 00      JM      OK
00E0      3C              INR     A
00E1      4F              MOV     C,A
00E2      11 80 F5      LXI     D,365:200Q
00E5      FE 01              CPI     1
00E7      C2 F2 00      JNZ     OK+3
00EA      C5              PUSH    8
00EB      CD 0C E0      CALL    SEEK
00EE      C1              POP     8
00EF      11 80 02      OK      LXI     D,2:200Q
00F2      2A 74 E2      LHL    DMAADDR
00F5      19              DAO     0
00F6      22 74 E2      SHLD    DMAADDR
00F9      3E 3F              MVI     A,77Q
00FB      94              SUB     H
00FC      FA 07 00      JM      NSOMA
00FF      C9              RET
    
```

TYPE B:TTCBIOS.PRM

```

;CBIOS DRIVERS FOR CP/M
;
2900 =      CPM      EQU      2900H
3106 =      ENTRY   EQU      CPM+806H
0004 =      CDISK   EQU      4
E000 =      ORIGIN  EQU      0E000H
E003 =      INPUT   EQU      ORIGIN+3
E006 =      OUTPUT  EQU      ORIGIN+6
E009 =      TKZERO  EQU      ORIGIN+9H
E00C =      TSEEK   EQU      ORIGIN+0CH
E00F =      SECTOR  EQU      ORIGIN+0FH
E012 =      DMA     EQU      ORIGIN+12H
E015 =      DISKR   EQU      ORIGIN+15H
E018 =      DISKW   EQU      ORIGIN+18H
E26E =      STACK   EQU      ORIGIN+26EH
E273 =      DRIVE   EQU      ORIGIN+273H
E274 =      DMAADR  EQU      ORIGIN+274H
E277 =      TRACK   EQU      ORIGIN+277H
E302 =      STATUS  EQU      ORIGIN+302H
E1F3 =      DELAY   EQU      ORIGIN+1F3H
E270 =      SCON    EQU      ORIGIN+270H
0003 =      IOBYTE  EQU      3H
0000 =      INTIOBY EQU      0
0080 =      RDYMSK  EQU      80H
0001 =      WPTCT   EQU      1
;
;
3E00      ;
;
3E00 C32D3E  START  JMP      BOOT
3E03 C36D7E          JMP      WBOOT
3E06 C3973E          JMP      CONST
3E09 C3C53E          JMP      CONIN
3E0C C3D73E  CPOUT  JMP      CONOUT
3E0F C3F23E          JMP      LIST
3E12 C3E73E          JMP      PUNCH
3E15 C3DD3E          JMP      READER
3E18 C3913E          JMP      HOME
3E18 C3343F          JMP      SELDISK
3E1E C3BD3E          JMP      SETTRK
3E21 C3823F          JMP      SETSEC
3E24 C312E0          JMP      DMA
3E27 C38C3F          JMP      READ
3E2A C3A13F          JMP      WRITE

```

```

3E2D 316EE2  BOOT  LXI      SP,STACK
3E30 CD613E  CALL     TINIT
3E33 21CC3F  LXI      H,PROMPT
3E36 CD8F3F  CALL     MESSG
3E39 AF      XRA      A
3E3A 320400  STA      CDISK
3E3D 32813F  STA      CDISKA
3E40 218000  GOCPM   LXI      H,80H
3E43 2274E2  SHLD    DMAADR
3E46 3EC3    MVI      A,OC3H
3E48 320000  STA      0
3E4B 21033E  LXI      H,START+3
3E4E 220100  SHLD    1
3E51 320500  STA      5
3E54 210631  LXI      H,ENTRY
3E57 220600  SHLD    6
3E5A 3A0400  LDA      CDISK
3E5D 4F      MOV      C,A
3E5E C30029  JMP      CPM
;
3E61 3E00    TINIT  MVI      A,INTIOBY
3E63 320300  STA      IOBYTE
3E66 210000  LXI      H,0
3E69 210000  LXI      H,0
3E6C C9      RET
;
3E6D 316EE2  WBOOT  LXI      SP,STACK
3E70 018000  LXI      B,200Q
3E73 CD12E0  CALL     DMA
3E76 3E8B    MVI      A,210Q
3E78 3273E2  STA      DRIVE
3E7B CD2AE0  CALL     ORIGIN+52Q
3E7E 214Q3E  LXI      H,GOCPM
3E81 229200  SHLD    222Q
3E84 3E2A    MVI      A,52Q
3E86 32CF00  STA      317Q
3E89 3E3D    MVI      A,(CPM+1400H)/100H
3E8B 32FA00  STA      372Q
3E8E C38000  JMP      200Q
;
3E91 CD393F  HOME  CALL     SELDSK
3E94 C309E0  JMP      TKZERO

```



3E97 0604	CONST	MVI	B,4	3EF2 210D3F	LIST	LXI	H,LTBLE
3E99 3A02E3		LDA	STATUS	3EF5 3A0300		LDA	IOBYTE
3E9C 1F		RAR		3EF8 1F		RAR	
3E9D 1F		RAR		3EF9 1F		RAR	
3E9E D2A73E		JNC	STATUS1	3EFA C3ED3E		JMP	PNCH1
3EA1 05		DCR	B				
3EA2 C2993E		JNZ	CONST+2	3EFD 293F	;	CITBLE	DW CITY
3EA5 AF		XRA	A	3EFF 293F			DW CICRT
3EA6 C9		RET		3F01 DD3E			DW READER
3EA7 CDO3E0	STATUS1	CALL	INPUT	3F03 293F			DW CIUC1
3EAA FE13		CPI	19	3F05 253F		COTBLE	DW COTTY
3EAC C2B13E		JNZ	STATUS4	3F07 253F			DW COCRT
3EAF AF		XRA	A	3F09 F23E			DW LIST
3EBO C9		RET		3F0B 253F			DW COUC1
3EB1 FE0D	STATUS4	CPI	ODH	3F0D 253F		LTBLE	DW COTTY
3EB3 C2A73E		JNZ	STATUS1	3F0F 253F			DW COCRT
3EB6 32BC3E		STA	STATUS5	3F11 253F			DW COLPT
3EB9 F6FF		ORI	OFFH	3F13 253F			DW COUL1
3EBB C9		RET		3F15 253F		PTBLE	DW COTTY
3EBC 00	STATUS5	DB	0	3F17 253F			DW COPTP
				3F19 253F			DW COUP1
				3F1B 253F			DW COUP2
3EBD C5	;	SETTRK	PUSH	3F1D 293F		RTBLE	DW CITY
3EBE CD393F		CALL	SELDSK	3F1F 293F			DW CIPTR
3EC1 C1		POP	B	3F21 293F			DW CIUR1
3EC2 C30CE0		JMP	TSEEK	3F23 293F			DW CIUR2
				3F25 =		COTTY	EQU \$
3EC5 21FD3E	CONIN	LXI	H,CITBLE	3F25 =		COCRT	EQU \$
3EC8 3A0300	CONIN1	LDA	IOBYTE	3F25 =		COUC1	EQU \$
3ECB 17		RAL		3F25 =		COLPT	EQU \$
3ECC E606	SELDEV	ANI	6H	3F25 =		COUL1	EQU \$
3ECE 1600		MVI	D,0	3F25 =		COPTP	EQU \$
3ED0 5F		MOV	E,A	3F25 =		COUP1	EQU \$
3ED1 19		DAD	D	3F25 =		COUP2	EQU \$
3ED2 7E		MOV	A,M	3F25 79			MOV A,C
3ED3 23		INX	H	3F26 C306E0			JMP OUTPUT
3ED4 66		MOV	H,M				
3ED5 6F		MOV	L,A				
3ED6 E9		PCHL		3F29 =		CITY	EQU \$
				3F29 =		CICRT	EQU \$
				3F29 =		CIPTR	EQU \$
3ED7 21053F	CONOUT	LXI	H,COTBLE	3F29 =		CIUC1	EQU \$
3EDA C3C83E		JMP	CONIN1	3F29 =		CIUR1	EQU \$
				3F29 =		CIUR2	EQU \$
3EDD 211D3F	;	READER	LXI H,RTBLE	3F29 21BC3E			LXI H,STATUS5
3EE0 3A0300		LDA	IOBYTE	3F2C 7E			MOV A,M
3EE3 1F	READR1	RAR		3F2D B7			ORA A
3EE4 C3CC3E		JMP	SELDEV	3F2E CA03E0			JZ INPUT.
				3F31 3600			MVI M,0
3EE7 21153F	;	PUNCH	LXI H,PTBLE	3F33 C9			RET
3EEA 3A0300		LDA	IOBYTE				
3EED 1F	PNCH1	RAR					
3EEE 1F		RAR		3F34 79		;	SELDISK MOV A,C
3EEF C3E33E		JMP	READR1	3F35 323D3F			STA NEWDRV+1
				3F38 C9			RET

```

3F39 21813F SELDSK LXI H,COISKA
3F3C 3E00 NEWDRV MVI A,0
3F3E BE CMP M
3F3F 77 MOV M,A
3F40 C8 RZ
3F41 4F MOV C,A
;
3F42 3A73E2 SELDSK1 LDA DRIVE
3F45 F5 PUSH PSW
3F46 E6F7 ANI OF7H
3F48 21793F LXI H,DTABLE
3F4B E5 PUSH H
3F4C BE SLOOP CMP M
3F4D CA553F JZ FOUND
3F50 23 INX H
3F51 23 INX H
3F52 C34C3F JMP SLOOP
3F55 23 FOUND INX H
3F56 3A77E2 LDA TRACK
3F59 77 MOV M,A
3F5A E1 POP H
3F5B 79 MOV A,C
3F5C 87 ADO A
3F5D 85 ADD L
3F5E 6F MOV L,A
3F5F D2633F JNC HISOK
3F62 24 INR H
3F63 F1 HISOK POP PSW
3F64 E608 ANI 8
3F66 B6 ORA M
3F67 3202E3 STA STATUS
3F6A 3273E2 STA DRIVE
3F6D 23 INX H
3F6E 7E MOV A,M
3F6F 3277E2 STA TRACK
3F72 AF XRA A
3F73 1102E3 LXI D,STATUS
3F76 C3F3E1 JMP DELAY
3F79 80 DTABLE DB 80H
3F7A DD DB 0
3F7B 01 DB 1
3F7C 00 DB 0
3F7D 20 DB 20H
3F7E 00 DB 0
3F7F 02 DB 2
3F80 0D DB 0
3F81 00 CDISKA DB 0
;
3F82 79 SETSEC MOV A,C
3F83 32883F STA SECTORA+1
3F86 C9 RET
;
3F87 0E00 SECTORA MVI C,0
3F89 C30FE0 JMP SECTOR
;

```

```

3F8C C0393F READ CALL SELOSK
3F8F CD873F CALL CALL SECTORA
3F92 DE0A MVI C,10
3F94 C5 REAO1 PUSH B
3F95 C015E0 CALL DISKR
3F98 C1 POP B
3F99 A7 ANA A
3F9A C8 RZ
3F9B 0D DCR C
3F9C C2943F JNZ READ1
3F9F 2F READY CMA
3FA0 C9 RET
;
3FA1 C0393F WRITE CALL SELDSK
3FA4 CD873F CALL CALL SECTORA
3FA7 0E0A MVI C,10
3FA9 C5 OSKW PUSH B
3FAA CD18E0 CALL DISKW
3FAD C1 POP B
3FAE A7 ANA A
3FAF C8 RZ
3FB0 D0 OCR C
3FB1 C2A93F JNZ OSKW
3FB4 3A02E3 PROTCT LDA STATUS
3FB7 E601 ANI WPTCT
3FB9 CA9F3F JZ READY
3FBC 21E23F LXI H,PTCTMSG
3FBF 7E MESSG MOV A,M
3FC0 A7 ANA A
3FC1 C8 RZ
3FC2 E5 PUSH H
3FC3 4F MOV C,A
3FC4 C00C3E CALL CPOUT
3FC7 E1 POP H
3FC8 23 INX H
3FC9 C3BF3F JMP MESSG
;
3FCC PROMPT OS 0
3FCC 000A DB 00H,0AH
3FCE 31364B2043 OB '16K CP/M'
3FD7 5645525320 DB 'VERS '
3FDC 312E34 DB '1.4'
3F0F 000A DB ODH,0AH
3FE1 00 DB 0
3FE2 000A PTCTMSG DB 00H,0AH
3FE4 50524F5445 DB 'PROTECT'
3FEB 0DOA DB 00H,0AH
3FEO 00 DB 0

```

A>

## HARDWARE LEVEL REGISTERS

Users desiring a greater level of control over the floppy disk or serial interface may wish to directly refer to the I/O device registers on the DJ from their 8080 program. There are eight one-byte registers, four of them are read only, and four of them are write only. The registers have four memory addresses on the S-100 bus with a different register being selected during a read operation and during a write operation. This gives the total of eight registers.

To make use of the registers, the first operation is to write to the write function register a bit pattern that selects one of the disk drives of the floppy disk system. All the other command and status bits will then refer to the operation of this selected drive.

### Readable Registers

Register 0 - Disk Data Register (location 343:000Q or E300H standard version):

Reading from this register will transfer one byte of information from the floppy disk to the CPU. Note that the correct use of this register requires programming sequences involving setting bits in other control registers and adhering to certain timing rules, or the data read will be meaningless.

Register 1 - Disk Read Mark Register (loc. 343:001Q or E301H standard version):

Reading from this register puts the CPU into a wait state until the DJ detects a field it recognizes as an address mark on the floppy disk. (See floppy disk data formats.) The bit pattern of the mark then appears as the contents of the register, and the CPU is allowed to leave the wait state and read in the register value.

If no valid data is found on the disk, the CPU will hang and an external reset will be necessary. It is therefore important for a disk to be in the drive, the door closed, the drive selected, and the head loaded and positioned to a valid track before a read mark is done.

Register 2 - Read Status Register (loc. 343:002Q or E302H standard version):

This register contains bits that identify the current status of the DJ and the currently selected drive.

BIT	7	6	5	4	3	2	1	0
	READY	SECTOR	2SIDED	INDEX	TRK00	HD-LOADED	SER-IN	WRT-PROT
	*	*	*	*	*			*

\*Bits marked with an asterisk reflect the current state of status lines from the floppy disk drive. See the documentation for the floppy disk drive actually in use for a detailed specification of these signals.

## Hardware Level Registers

- READY - This bit is a 1 when the currently selected disk drive is powered on, the door is closed, and a diskette is present.
- SECTOR - This line reflects the status of the sector status line from the floppy disk drive.
- 2SIDED - This line is a 1 when the status line from the floppy disk drive indicates that it is equipped to read two-sided floppy disks.
- INDEX - This line reflects the status of the INDEX line from the floppy disk drive. It goes to a 1 once per revolution of the floppy disk.
- TRK00 - This bit is a 1 when the read/write head in the floppy disk is positioned on track 00.
- HD-LOADED - This bit is a 1 when the DJ is directing the floppy disk drive to press the read/write head against the recording medium.
- SER-IN - This bit reflects the current status of the TTY or RS-232 serial input line.
- WRT-PROT - This bit is a 1 when the diskette in the drive is write protected, that is, there is a notch in the jacket for 8" drives or there is no notch for 5" drives.

Register 3 - Load Head Register (loc. 340:003Q or E303H standard version):

Reading a byte of data from the address of this register causes the currently selected floppy disk drive to load the read/write head, that is, operate the solenoid to bring the head into contact with the medium. The byte of data that will actually be read from this memory location is meaningless. The head will remain loaded for sixteen revolutions after a load head command and will then automatically unload unless bit 3 of the write function register is a 0.

## Write Only Registers

Register 0 - Disk Data Register (loc. 340:000Q or E300H standard version):

The byte of data to be written to the disk is written to this register. As with the Read Disk Data Register, a proper sequence of setup commands is necessary for a write to this register to be effective.

## Hardware Level Registers

Register 1 - Disk Write Mark Register (loc. 340:001Q or E301H standard vers.):

Writing a byte of data to this register, when done at the proper time and place in a programming sequence, can write one of the special synchronization marks onto the disk. These synchronization marks are recorded on the disk in a way that is immediately distinguishable from data since certain clock pulses are omitted in accordance with industry standards. The DJ automatically generates the correct pattern of missing clock pulses to be recorded with the mark written.

The standard marks and the hex value of the byte of data that should be written to this register to get them are:

FB Data Mark  
F8 Deleted Data Mark (rarely used)  
FE Sector Mark (only used during initialization of disk)  
FC Index Mark (only used during initialization of disk)

Register 2 - Write Function Register (loc. 340:002Q or E302H standard vers.):

This register contains bits that activate various control signals to the floppy disk drive(s) and to the internal circuitry of the DJ.

BIT	7	6	5	4	3	2	1	0
	DS1	IN-USE	DS3	STEP	E-A-U	DIR	DS4	DS2

DS1-DS4 - These bits are passed on to the output flat cable to the floppy disk drives as DRIVE SELECT 1 thru DRIVE SELECT 4. Each disk drive has a means of setting an internal jumper to determine which drive select line in the cable it will respond to. Only the disk drive whose drive select line is on will respond to the commands output by the program to the writable registers and present drive status in the readable registers. Likewise, only the selected drive will participate in data transfers. Only one drive should be selected at a time.

IN-USE - This line is normally not used. Setting it to a 1 forces the LED on the front of the Shugart compatible floppy disk drive to be on regardless of the state of other activity at the drive.

STEP - This bit controls the step line to the floppy disk drive. Setting this bit from a 1 to a 0 causes the stepping motor in the floppy disk drive to move the read/write head one track in the direction specified by the DIR bit.

## Hardware Level Registers

DIR - This bit indicates to the floppy disk drive the direction it should move the read/write head in response to a step command:

0 = out  
1 = in

E-A-U Enable Auto Unload. This bit being a 1 activates the automatic head unload feature. To prolong medium life, it is important to minimize the amount of time the read/write head is pressed against the disk. When this bit is set to a 1, the DJ will automatically unload the read/write head sixteen revolutions after the last load head command was issued. This bit should always be set to a 1 except for hardware debugging or data transfer optimization purposes.

Register 3 - Write Serial Register (loc. 340:003Q or E303H standard version):

Bit 5 of this register will appear at an appropriate voltage (or current) level at the RS-232 (and TTY) serial output connector. The other bits of this register have no significance.

## PARTS LIST

7	180Ω ¼ watt resistors	(brown-grey-brown)	R1, 8, 9, 10, 11, 12, 13
3	240Ω ½ watt resistors	(red-yellow-brown)	R21, 27, 28
1	470Ω 1 watt resistor	(yellow-purple-brown)	R20
4	510Ω ¼ watt resistors	(green-brown-brown)	R22, 23, 25, 26
4	1kΩ "	(brown-black-red)	R3, 16, 17, 24
1	1.5kΩ "	(brown-green-red)	R14
3	3.3kΩ "	(orange-orange-red)	R4, 7, 15
1	3.9kΩ "	(orange-white-red)	R2
1	4.7kΩ "	(yellow-purple-red)	R19
2	27kΩ "	(red-purple-orange)	R5, 6
1	47kΩ "	(yellow-purple-orange)	R18
1	.01 μfd disk capacitor		C1
17	by-pass capacitors*		
1	.82 μfd tantulum capacitor		C3
2	2.7 μfd "		C5, 6
2	39 μfd "		C2, 4
1	1N914/4820-0201 signal diode		CR1
2	1N4742 12 volt zener diodes		CR2, 3
1	2N3904 NPN transistor		Q2
2	2N3906 PNP transistors		Q1, 3
2	slide switches		
1	heat sink		
1	set machine nut and screw		
1	4Mhz crystal		
1	50 pin right angle flat cable connector		J1
1	7 pin right angle connector w/ matching header and pins		J2

## Parts List

1	8 pin low profile socket	
16	14 pin low profile sockets	
14	16 pin low profile sockets	
5	20 pin low profile sockets	
1	74LS00 quad 2-input NAND gate	3C
1	74LS02 quad 2-input NOR gate	10B
1	74LS08 quad 2-input AND gate	5B
1	74LS10 tri 3-input NAND gate	3B
1	74LS14 hex Schmitt-trigger inverter	5C
2	74LS20 dual 4-input NAND gate	12C, 13A
1	74LS32 quad 2-input OR gate	8B
8	74LS74 dual D-type flip flop	9A, 10A, 1B, 2B, 12B, 13B, 1C, 2C
1	74LS124 dual voltage-controlled oscillator	8A
1	74LS155 dual 1 of 4 decoder	4C
2	74LS161 synchronous 4-bit counter	4B, 13C
1	74165 parallel-load 8-bit shift register	9B
1	74LS174 hex D-type latch with clear	7A
1	74LS241 octal tri-state** bus driver	10C
1	74LS273 octal D-type latch with clear	5A
1	74LS299 8-bit bidirectional shift/storage register	11C
2	74366/368 hex 2-4 tri-state** inverting bus drivers	3A, 6A
1	DM8090 quad inverter, dual 2-input NAND gate	11B
1	DM81LS95/97 octal tri-state** buffer	9C
1	MMI 6301/82S129 4x256 bit tri-state** PROM	8C
2	MMI 6306/82S131 4x512 bit tri-state** PROM	6C, 7C
1	DM81LS96/98 octal tri-state** inverting buffer	4A



## Parts List

2	2112	4x256 bit RAM with tri-state** output	6B, 7B
1	7805	5 volt 1 amp monolithic regulator	1A
1	741	operation amplifier	11A
1		5" x 10" printed circuit board	
1		glossy photograph	

\*by-pass capacitors will vary in value from .01  $\mu$ fd to .1  $\mu$ fd depending upon currently available supplies.

\*\*tri-state is a trademark of National Semiconductor Corp.

## ASSEMBLY INSTRUCTIONS

DO NOT INSTALL OR SOLDER ANY PARTS UNTIL YOU HAVE READ THESE INSTRUCTIONS SEVERAL TIMES AND HAVE FULLY DIGESTED THE INFORMATION!

CAUTION -- DO NOT SOLDER OR CLIP COMPONENT LEADS WITHOUT USING SAFETY GLASSES!

### INSPECTION

Use the Parts List to make sure that there are no missing items in your kit. Please notify us of any shortages. Be sure to check for missing parts before you start to assemble.

### COMPONENT LEAD WIDTHS

Bend the leads with the plastic bending block in your kit. Be sure that the leads are bent to the proper width before the part is inserted. Properly bent components solder easier and give the finished kit a professional appearance. See the right hand column of the Parts Installation for proper component widths.

### SOCKETS

A socket is furnished for every integrated circuit. It is important that you use the sockets; otherwise, a defective part will be extremely difficult to replace.

NO REPAIR OR SERVICE WILL BE PERFORMED ON A KIT WHICH HAS HAD INTEGRATED CIRCUITS SOLDERED TO THE CIRCUIT BOARD.

### PARTS ORIENTATION

In all references throughout the instructions, the convention used is that the gold edge connector is the bottom of the board. Orientation identification is molded into the plastic of the sockets and is illustrated below:

## Assembly Instructions

This orientation mark identifies where pin #1 of the integrated circuit is to be positioned when inserted into the socket. The socket should be inserted in the board so that the orientation mark is in the lower right hand corner.

Orientation of the transistors, tantalum capacitors, diodes and voltage regulator is specified in the component layout drawing. It is advisable to study this drawing and the 8 x 10 glossy photograph carefully before building the kit. Refer to both during parts installation.

### SOLDERING AND SOLDER IRONS

The most desirable soldering iron for complex electronic kits is a constant temperature soldering iron with an element regulated at 650<sup>0</sup> F. The tip should be fine so that it can be brought in intimate contact with the pads of the circuit board. Both Unger and Weller have excellent products which fit the above requirements.

There are three important soldering requirements for building this kit.

1. Do not use an iron that is too cold (less than 600<sup>0</sup> F) or too hot (more than 750<sup>0</sup> F).
2. Do not apply the iron to a pad for extended periods.
3. Do not apply excessive amounts of solder.

The proper procedure for soldering components to the circuit board is as follows:

1. Bring the iron in contact with both the component lead and the pad.
2. Apply a small amount of solder at the point where the iron, component lead, and pad all make contact.
3. After the initial application of solder has been accomplished with the solder flowing to the pad and component lead, the heat of the iron will have transferred to both the pad and the lead. Apply a small amount of additional solder to cover the joint between the pad and the lead. **DO NOT PILE SOLDER ON THE JOINT! EXCESSIVE HEAT AND SOLDER CAUSE PADS AND LEADS TO LEFT FROM THE CIRCUIT BOARD. EXCESSIVE SOLDER IS THE PRIMARY CAUSE FOR BOARD SHORTS AND BRIDGED CONNECTIONS.**

## Assembly Instructions

### PARTS INSTALLATION

Before installing parts, bend the leads of the resistors, diodes, and tantulum capacitors to the lengths shown in the right hand column. After a series of parts have been installed in the board, bend the leads slightly to hold them in place, solder the leads and trim the excess lead lengths before proceeding to the next step.

#### Install:

CR1	1N914/4820-0201 signal diode	Check for orientation!	.5 in.
R1	180 $\Omega$ $\frac{1}{4}$ watt resistor	brown-grey-brown	.5
R2	3.9k $\Omega$ "	orange-white-red	.5
R3	1k $\Omega$ "	brown-black-red	.5
R4, R7	3.3k $\Omega$ "	orange-orange-red	.5
R5, R6	27k $\Omega$ "	red-purple-orange	.5
R8-R13	180 $\Omega$ "	brown-grey-brown	.5
R22	510 $\Omega$ "	green-brown-brown	.5
R14	1.5k $\Omega$ "	brown-green-red	.5
R15	3.3k $\Omega$ "	orange-orange-red	.5
R16, R17	1k $\Omega$ "	brown-black-red	.5
R18	47k $\Omega$ "	yellow-purple-orange	.5
R19	4.7k $\Omega$ "	Yellow-purple-red	.5
R23	510 $\Omega$ "	green-brown-brown	.5
R24	1k $\Omega$ "	brown-black-red	.5
R25, R26	510 $\Omega$ "	green-brown-brown	.5
CR2, CR3	1N4742 12 v. zener diode	Check for orientation!	.5
C3	.82 $\mu$ fd tantulum capacitor	"	.5
C5, C6	2.7 $\mu$ fd "	"	.5
R21, 27, 28	240 $\Omega$ $\frac{1}{2}$ watt resistor	red-yellow-brown	.6
4 Mhz crystal		Use a trimmed resistor lead to secure the crystal to the circuit board	.45

## Assembly Instructions

- C2, C4      39  $\mu$ fd tantulum capacitor      Check for orientation!    .6 in.
- Socket 11A    8-pin low profile      "
- Sockets 3A, 6A-8A    16-pin low profile      "
- Sockets 9A, 10A, 13A    14-pin low profile      "
- Sockets 4A, 5A    20-pin low profile      "
- Sockets 1B-3B, 5B, 8B, 10B, 12B, 13B  
                  14-pin low profile      "
- Sockets 4B, 6B, 7B, 9B, 11B    16-pin low profile      "
- Sockets 1C-3C, 5C, 12C    14-pin low profile      "
- Sockets 4C, 6C-8C, 13C    16-pin low profile      "
- Sockets 9C-11C    20-pin low profile      "
- R20            470 $\Omega$  1 watt resistor      yellow-purple-brown      .8
- J2            7-pin right angle header with  
                  pins oriented toward the top of  
                  the board (away from the 100 pin  
                  edge connector)      Check for orientation!
- J1            50-pin right angle header with  
                  pins oriented toward the top of  
                  the board (away from the 100  
                  pin edge connector)      "
- Slide switches S1 and S2  
                  After installation, these two switches should be  
                  switched in the same direction: to the left if the  
                  Disk Jockey is to be used with a full-size floppy  
                  disk or to the right if it is to control a 5" mini-  
                  floppy disk drive.
- C1            .01  $\mu$ fd disk capacitor
- By-pass capacitors (17 each). Values may vary from .01  $\mu$ fd to  
                  .1  $\mu$ fd.
- LM340.5/7805 5 volt regulator  
                  Bend lead, insert and hand tighten the nut and bolt  
                  with the bolt running through the bottom of the board  
                  through the heat sink and through the regulator. The  
                  nut should be hand tightened over the regulator. Solder  
                  the leads. If heat sink grease is available, apply a  
                  thin film between the board, heat sink, and regulator.  
                  Finally, tighten the nut firmly.

## POWER-UP AND SYSTEM CHECK OUT

### Power Supply/Voltage Regulator Check Out

Voltage requirements: (reference to ground - pins 50 and 100)

Pins 1 and 51	no less than 7 volts not more than 10 volts	approx. 700 ma
Pin 2	not less than 13 volts not more than 22 volts	approx. 25 ma
Pin 52	not less than -22 volts not more than -13 volts	approx. 100 ma

Before installing any of the integrated circuits, apply power to pins 1 and 51, pin 2 and pin 52 (ground at pins 50 and 100) as specified above. Perform the following measurements with a volt meter:

(1) Pin 4 of 11A (741)	-12 volts
(2) Pin 7 of 11A	+12 volts
(3) Pin 14 of 13A	+5 volts
(4) Pin 14 of 13B	+5 volts
(5) Pin 16 of 13C	+5 volts

If the voltage at any of the check points differs from the required value, return the unit for trouble shooting and repair.

### Power-Up Check Out

Install the integrated circuits as per the layout on the board (note: IF the legend at 5A reads 74LS373, change to 74LS273). When inserting these parts, be careful not to bend pins under the package -- a pin which is bent under the integrated circuit may appear to be inserted in the socket. **BENT PINS ARE THE MOST COMMON REASON FOR MALFUNCTIONING BOARDS!**

After all the parts have been installed, voltages checked, and integrated circuits installed, reconnect the power supplies and power up the board again. This can be done stand-alone or in the mainframe of your computer. Check again the voltages called out in the previous section. If there are any differences from the required values, please return the board.

### 8"/5" Hardware Selection

Be sure that the two slide switches S1 and S2 are to the left for 8" drives and to the right for 5" drives. **BOTH SWITCHES MUST FACE THE SAME DIRECTION -- BOTH TO THE LEFT OR BOTH TO THE RIGHT.**

## Power-Up and System Check Out

### SYSTEM CHECK OUT

#### Introduction

The Disk Jockey interface occupies 1024 bytes of memory 340:000Q through 343:377Q (E000H through E3FFH). These addresses can be customized for special applications but in the check out procedures which follow, standard addressing will be assumed. Users of customized boards should apply appropriate address offsets.

#### ROM

(1) Examine location 340:000Q and verify that it contains 303Q (location E000H should contain C3).

(2) Examine location 340:003Q and verify that 303Q is also present (E003 should contain C3).

(3) Examine location 341:377Q and verify that it contains 311Q (E1FF should contain C9).

#### RAM

(1) Examine location 342:000Q (E200H).

(2) Verify that the following values can be deposited at the stated addresses.

Octal		Hex	
Address	Value	Address	Value
342:000	000	E200	00
342:001	001	E201	01
342:002	002	E202	02
342:003	004	E203	04
342:004	010	E204	08
342:005	020	E205	10
342:006	040	E206	20
342:007	100	E207	40
342:010	200	E208	80
342:011	377	E209	FF

## Power-Up and System Check Out

### I/O

#### Disk Pause Logic

Enter the following program in the Disk Jockey RAM:

	Octal		Hex					
342:000	072 001 343	E200	3A 01 E3	LOOP	LDA	MARK		
342:003	303 000 342	E203	C3 00 E2		JMP	LOOP		

Start this program at the label LOOP. The system should hang with PREADY low. Turn the computer off and then on in order to generate a  $\overline{POC}$  (power-on-reset). Note: many S-100 systems generate a POC signal when the reset switch is pressed and in this case it is not necessary to turn the system off and then on again. In the sequel, the above procedure will be referred to as "generate a  $\overline{POC}$  signal."

Enter the following program in the Disk Jockey RAM:

	Octal		Hex					
342:000	072 000 343	E200	3A 00 E3	LOOP	LDA	DATA		
342:003	303 000 342	E203	C3 00 E2		JMP	LOOP		

Start the program. This time the system should not hang. Stop the computer. This completes the test of the disk pause logic.

#### Load Head Command

Generate a  $\overline{POC}$  signal. Verify that pins 1 and 15 of IC 6A are at a logic 1 (approx. 4-5 volts). Enter the following program in the Disk Jockey RAM.

	Octal		Hex					
342:000	072 003 343	E200	3A 03 E3	LOOP	LDA	HEAD		
342:003	303 000 342	E203	C3 00 E2		JMP	LOOP		

Start this program and after a moment, stop the program. Verify that pins 1 and 15 of IC 6A are now at a logic zero (approx. 0 volts). For the next test, it is necessary that pins 1 and 15 are at a logic zero. This completes the test of the Load Head Command.

#### Disk Function Register

This test is to be performed just after the Load Head Command test. Before the Disk Function Register is tested, pins 1 and 15 of IC 6A should be at a logic zero level. Enter the following program:

342:000	062 011 342	E200	32 09 E2	LOOP	LDA	TEST		
342:003	072 002 343	E203	3A 02 E3		STA	FUNCTION		
343:006	303 000 342	E206	C3 00 E2		JMP	LOOP		
343:011	XXX	E209	XX	TEST	DATA			



## Power-Up and System Check Out

The above program is to be run with various values for DATA deposited in location TEST. With each value of DATA, pins 16, 26, 28, 30, 32, 34 and 36 of J1 are to be probed. For each different value of DATA, exactly one of these pins should be at a logic zero and all the rest at a logic one. The table below details what the different values for DATA should be and what pin of J1 should be and what pin of J1 should be at logic zero for the given value of DATA.

Value Deposited at location TEST		J1 Pin Which Should be at Logic Zero
Octal	Hex	
001	01	28
002	02	32
004	04	34
020	10	36
040	20	30
100	40	16
200	80	26

Each time the value of TEST is changed, the short program at the beginning of this section should be run. While it is running, the various pins of J1 detailed above should be probed. After the pins have been checked, the program should be stopped and the next value in the table should be deposited at TEST, etc. This completes the test of the disk function register.

### Disk Write Logic

If a logic probe or oscilloscope is available, attach it to pin 38 of J1. Momentarily ground pin 1 of IC 13B. There should be a stream of negative pulses 250 ns wide, four microseconds apart. In any event, pin 1 of IC 3C should be at a logic 1 as well as pin 40 of J1. Enter the following program in the Disk Jockey RAM:

Octal	Hex	
342:000 041 000 343	E200 21 00 E3	LOOP LXI H,DATA
342:003 066 377	E203 36 FF	MVI M,377Q (FFH)
342:005 303 003 342	E205 C3 03 E2	JMP LOOP+3

Start the program. While the program is running, verify that pin 40 of J1 is now at a logic 0. If a logic probe or oscilloscope is available, verify that pin 38 of J1 has a stream of negative pulses 250 ns wide, two microseconds apart. Also verify that pin 1 of IC 3C has a stream of negative pulses 250 ns wide, four microseconds apart. Stop the program. This completes the test of the disk write logic.

## Power-Up and System Check Out

### Disk Status Register

Generate a  $\overline{POC}$  signal. Read the following list of values from location 343:002Q (E302H). Associated with each value is a pin # of J1. In each case, ground only this pin while examining 343:002Q.

Value at 343:002Q		Pin Number of J1 to be Grounded
Octal	Hex	
001	01	44
010	08	42
020	10	20
040	20	10
100	40	24
200	80	22

### Serial Input Port

Generate a  $\overline{POC}$  signal. With a jumper connect J2 pin 1 to 11A pin 4. Examine location 343:002Q with this jumper in place and verify that 343:002 now contains the value 2. Disconnect the jumper. Examine location 343:003. Verify that the value of 343:002 is zero. This completes the test of the serial input port.

### Serial Output Port

Generate a  $\overline{POC}$  signal. Verify that pin 2 of J2 is at or near -12 volts. Enter the following program in the Disk Jockey RAM:

	Octal		Hex						
342:000	072	011	342	E200	3A	09	E2	LOOP	LDA TEST
342:003	062	003	343	E203	32	03	E3		STA SERIAL
342:006	303	000	342	E206	C3	00	E2		JMP LOOP
342:011	XXX			E209	XX			TEST	DATA

DATA is to have two values: 40Q (20H) and zero. Run the program with DATA = 0 and verify that pin 2 of J2 is at or near +12 volts. When DATA = 40Q, pin 2 of Jw should be at or near -12 volts.

This completes the preliminary check out of the Disk Jockey controller. Further check out will require that a drive be connected to the board.

### CABLE ORIENTATION

The cable should be so that the ribbon cable leaves the connector toward the back of the computer and away from the component side of the Disk Jockey circuit board. Care should be exercised in fabricating a disk cable. The connectors at the cable ends must be oriented so that pin 2 of J1 matches pin 2 of the drive connector. A cable for Shugart drives is available from Thinker Toys.

## Power-Up and System Check Out

### DISK DRIVE CHECK OUT

The following tests can be conducted after the appropriate disk drive is connected to the Disk Jockey (via the 50 conductor cable) and proper power is furnished to the drive.

#### Track Zero Seek

Set up the disk drive so that it is drive #1. Make sure that the head is away from the track zero sensor. Enter the following program into the memory of the host computer.

Octal	Hex		
000:076 210	00 3E 88	MVI	A,210Q
002:072 163 342	02 3A 73 E2	STA	DRIVE
005:072 002 343	05 3A 02 E3	STA	FUNCTN
010:061 150 342	08 31 58 E2	LXI	SP,STACK
013:315 011 340	0B CD 09 E0	CALL	TKZERO
016:303 016 000	0E C3 0E 00	JMP	STALL

Insert a write protected diskette into the drive and close the door securely. Start the program. The head should load and move outward until it reaches track 00. After sixteen revolutions, the head should unload. Do not turn off power to the computer and drive between this test and the next. Do not change the stack pointer.

#### Seek and Disk Read

In the next test, be sure that a write protected disk which is cleanly initialized is inserted in the drive. Enter the following program:

Octal				
100 016 002	START	MVI	C,2	SEEK
102 315 014 340		CALL	TSEEK	TRACK 2
105 016 001		MVI	C,1	SET
107 315 017 340		CALL	SECTOR	SECTOR 1
112 041 000 001		LXI	H,1:000	SET DMA
115 042 164 342		SHLD	DMA	TO PAGE 1
120 315 025 340		CALL	DISKR	READ DATA
123 303 123 000	STOP	JMP	STOP	

Hex				
40 0E 02	START	MVI	C,2	
42 CD 0C E0		CALL	TSEEK	
45 0E 01		MVI	C,1	
47 CD 0F E0		CALL	SECTOR	
4A 21 00 01		LXI	H,100	
4D 22 74 E2		SHLD	DMA	
50 CD 15 E0		CALL	DISKR	
53 C3 53 00	STOP	JMP	STOP	

## Power-Up and System Check Out

Start the program. The head should load and move to track 2 and unload after sixteen revolutions. Verify that the value 345 is contained in all memory locations between 1:000Q and 1:177Q.

This completes the check out of the Disk Jockey controller.

### Power Supply

The board has a worst case requirement for

800 ma of +8V (between 7.5V and 11V)  
100 ma of -16V (between -12.5V and -20V)  
30 ma of +16V (between +12.5V and +20V)

Plus 5 volt power (Vcc) is derived from the +8V supply in the computer backplane by means of the 5 volt 7805/LM340.5 regulator with C2 and the small disk capacitors labeled on the circuit board legend providing power supply bypass capacitance.

The +12V supply is derived from the backplane +16V supply by zener CR3 and R28, filtered by C6.

The -12V supply is derived similarly from the backplane -16V supply by means of CR2, R27, and C5.

### Floppy Disk Section System Clock

The basic timing for the DJ is derived from a crystal stabilized 74LS124 oscillating at 4 MHz. This clock signal is divided by two to produce 2 MHz SYS CLK frequency appropriate for a standard floppy at pin 9A-5. This signal is divided by two again to produce the 1 MHz SYS CLK appropriate for a mini-floppy at pin 9A-9. The appropriate frequency is selected by switch SW2 to drive the SYS CLK line. An inverted version of the system clock is generated at pin 10B-10.

### Device Address Decoding

The Disk Jockey contains 256 bytes of RAM, 512 bytes of ROM, and I/O registers. The memory address where each of these modules appears is controlled by the MMI6301 256x4 PROM in slot 8C. This PROM monitors the highorder address lines A8-A15. Connections to the chip select pins of SINTA, SINP, and SOUT make sure that the PROM is only active when the address present on the S-100 bus is a valid memory address during either a bus memory read or write cycle.

Take, as an example, the RAM ENBL output of the PROM. Normally high, this output should go low when the address on A8-A15 corresponds to the high order address byte that should select the RAM. The PROM internally has 256 cells, each one selected by a different combination of the high order address bits. All of these cells have 1 stored in them except for the one corresponding to the address where the RAM should be located which has a 0 stored in it. Thus, when the CPU refers to any location within the 256 byte page of memory where the RAM should be located, the cell with the zero in it is selected by A8-A15 and the RAM ENBL output goes low, selecting the RAM (see System RAM below).

## Hardware Description

The firmware ROM, being 512x8, has two consecutively addressed bits of zero in the address decoding ROM pattern. Thus, the signal ROM ENBL is generated for either 256 byte page of ROM. Address line A8 also goes to these ROMs to allow them to decide which 256 byte page within them is being called for (see System ROM below).

Likewise, the signal I/O ENBL is generated when the CPU refers to memory addresses within the 256 byte page assigned to the I/O device registers.

The fourth output, 8C-9, is programmed to go low if any of the other outputs of the address decoding ROM is low, that is, if the CPU is referring to any address within the DJ.

### Internal Data Bus

Data flow in the floppy disk controller is organized around an eight-bit wide internal data bus DATA0 - DATA7.

When the CPU is writing to the RAM or an I/O register, the S-100 bus signal MWRITE is high or WR is low. The WRITE line (5B-11) therefore goes low. Data flows from the S-100 bus data out lines (D00 - D07) through the 81LS97 in slot 9C to the internal data bus. Depending on what DJ register is being loaded, other control logic causes the data to be loaded from the internal data bus to the appropriate destination.

When the CPU is reading from the ROM, RAM, or I/O registers, other logic, detailed below, causes the selected data byte to be gated onto the internal data bus. When the line PDBIN goes high, indicating that the CPU is reading and the PROM in slot 8C detects that the CPU is addressing a region of memory assigned to the DJ and therefore grounds pin 8C-9, the signal INPUT ENBL goes low. Since pin 10C-1 is now low and pin 10C-19 high, the 74LS241 in slot 10C gates the data byte from the internal data bus to the S-100 bus data lines (DI0 - D17).

### Processor Wait States

Whenever a reference is made to the I/O registers of the Disk Jockey, the line READY is grounded to add two wait states to the CPU's cycle. The disk data transfer operations can require up to two processor cycles before they can obtain synchronization and ground READY to indicate that they require wait states. Two wait cycles are therefore added to every address reference within the Disk Jockey I/O registers to make sure the CPU does not leave the wait state before the disk data transfer control logic can ground READY.

The two flip-flops in 1B-9 and 1B-5 act as a shift register to supply the two machine cycle delay required. These flip-flops are always cleared by PSYNC, connected in an inverted version to 1B-13 and 1B-1. 1B-5 will then stay low until the second negative edge of PHI-2 after PSYNC goes low. At this time, the high level connected to 1B-12 will appear at 1B-5 having been shifted through the two flip-flops. During this interval when

## Hardware Description

1B-5 is low, 3B-11 will be held low, so 3B-8 will be held high. This will happen whether or not the CPU is referencing the Disk Jockey. However, only if the CPU is referencing Disk Jockey I/O registers will the signal I/O ENBL be low, enabling the 74367 in slot 3A to drive PREADY low. If necessary, the other inputs to gate 3B (9 and 10) are driven low by the disk data transfer control logic before 3B-11 goes high to add additional wait states (see below).

Due to the high speed of the disk, the use of processor wait states is essential for synchronization. If a flag bit were provided which the processor could test to determine if the floppy disk was ready for data transfer and branch accordingly, the execution of this loop would take up too much time to allow transfer of data from memory to disk. The use of processor wait states, suggested by Eugene Fisher of Lawrence Livermore Laboratory, synchronizes the processor instruction execution to the disk data transfer and ensures sufficient time for each byte of data to be transferred from device registers to memory before the next byte must be transferred.

### System RAM

The Disk Jockey contains two 2112 256 word x 4 bit RAMs located in slots 6B and 7B. When the CPU during a read or write operation puts the address assigned to this RAM onto the high order S-100 bus address lines (A8-A15), RAM ENBL (8C-11) goes low. This signal activates the chip select pin (13) of both RAMs. Whether the RAM does a read or a write operation is controlled by the WRITE line which connects to the R/W pin (14) of both RAMs. During a write, data from the CPU flows from the internal data bus into the bidirectional I/O pins of the RAMs. During a read, data from the RAMs flows out onto the internal data bus and from there onto the S-100 bus.

### System ROM

The Disk Jockey is provided with two MMI6306 512 word x 4 bit ROMs in slots 6C and 7C. These ROMs contain 512 bytes of disk utility software. When the address of the area of memory assigned to the ROM appears on the S-100 bus address lines (A8-A15), the signal ROM ENBL (8C-12) goes low. This enables the two ROMs by grounding their CS pin (13). The nine low order S-100 bus address bits (A0 - A8) specify to the ROMs which word is to be read. The ROM drives the byte of software stored at this location onto the internal data bus.

### I/O Registers

When the address decoding PROM in slot 8C detects an S-100 bus memory reference to an address within the 256 byte page assigned to the I/O registers, it grounds 8C-10 (I/O ENBL). By grounding 4C-2 and 4C-14, this activates the 74LS155 decoder in slot 4C whose function is to decide which

## Hardware Description

of the eight I/O registers is being referred to. The four readable registers and the four writeable registers are each assigned a section of the decoder. The two low order S-100 bus address bits select one of the four registers in each group. If the CPU is engaged in a write operation, the line WRITE will be low and the selected output of the "write section" decoder will go low (4C-9, 10, 11 or 12). If the CPU is engaged in a read operation, the line INPUT ENBL will be high, and the selected output of the read section of the decoder will go low (4C-4, 5, 6, or 7). Note that the definitions of WRITE and INPUT ENBL are such that either a write register or a read register will be selected, never both.

The uses of the READ DATA, WRITE DATA, READ MARK, and WRITE MARK signals are discussed below under the disk data flow section.

The READ STATUS signal, when low, enables the 81LS97 octal buffer in slot 4A, allowing the status byte (six bits of status from the floppy disk drive cable, the TTL level serial input, and the status of the head flip-flop) to flow onto the internal data bus.

The LOAD HEAD signal, when low, resets the 74LS161 hex counter in slot 4B and sets the 74LS74 flip-flop in slot 10A. This flip-flop drives the LOAD HEAD signal to the disk drives and enables the 74368 driver in slot 6A. The 74368 driver at 6A drives the disk select lines over the flat cable to the disk drive(s).

The WRITE FNCTN signal, when low, clocks the contents of the internal data bus into the 74LS273 octal register in slot 5A. Seven of the data bits set control lines to the floppy disk and are driven onto the flat cable from the 74LS273 outputs by the 74368 drivers in slots 3A and 6A. Bit 3 of the octal latch controls whether the read/write head will automatically unload after sixteen revolutions of the disk. The latch output at 5A-9 is connected to 4B-7, the enable input of a 74LS161 counter. This counter is reset to zero when the heads are loaded. The flip-flop in slot 10A is also set to 1 enabling the 74368 to drive the current setting of the drive select lines and the load head signal onto the flat cable. If bit 3 specifies that automatic head unloading is not to take place, the counter is not enabled, and no counting takes place. If the bit enables the counter, then every time the index point of the disk is passed (assuming the correct setting of SW1), the pulse on 4B-2 will cause the counter to count up by 1. When the counter finally counts to 15, the carry output at 4B-15 will go high. This will clock the 74LS74 at 10A-11, which will load a 0 from the data input 10A-12. The head load flip-flop will therefore be reset, disabling the driver in slot 6A, deselecting all drives and causing the LOAD HEAD signal to become inactive. Note that there is no explicit command to unload the heads; it can only be done by this timeout mechanism.

The WRITE SERIAL signal clocks the 74LS74 serial out flip-flop at 10A-3 whose data input is bit 5 of the internal data bus. The flip-flop output 10A-5 drives the translator from TTL to RS232 levels for the serial output; 10A-6 controls the 20 ma TTY driver. These level translation circuits are discussed in more detail below.



## Serial Communications Level Translators

### Serial Input:

A terminal device is connected to either the RS-232 or the TTY input pin.

If the RS-232 input pin is at a "SPACE" level (greater than 3 and less than 12 volts), transistor Q2 is cut off, so transistor Q1 is cut off, and the SERIAL INPUT line is pulled up to Vcc through R19 and the string of R5, R6, and R20. When the Read Status Register is read, the SERIAL INPUT line is gated onto bit 1 of the internal data bus through the 81LS97 in slot 4A, and a 0 appears in bit 1 of the word read. If the RS-232 input pin is at the "MARK" level (less than -3 but greater than -12 volts), current flows through transistor Q2, causing Q1 to pull the SERIAL INPUT line to ground. A 1 will therefore appear in bit 1 of the register when it is read.

If current is allowed to flow between the TTY+ input, through the terminal device, and back into the TTY- input (-12 volt return), representing the "MARK" condition, the voltage level at the SERIAL INPUT line is pulled down to approximately ground through R5 and R6. If no current is flowing ("SPACE"), the voltage level of the SERIAL INPUT line is pulled up to Vcc through R20, R5, and R6 (as well as R19).

### Serial Output:

When the program writes a byte to the Write Serial Register, the signal WRITE SERIAL at pin 10A-3 clocks the value of bit 5 of the internal data bus into flip-flop 10A-5. The Q output 10A-5 is compared to a reference value of 1.6 volts developed by R15 and R14 at the 741 op amp. The output of the op amp will be saturated at approximately -12 volts for a "MARK" and approximately +12 volts for a "SPACE". This output drives the RS-232 level output through resistor R4. Capacitor C1 is required to lengthen the rise and fall time of the RS-232 output, as required by specifications.

Meanwhile, the  $\bar{Q}$  output 10A-6 is driving transistor Q3 which acts as a current source for the TTY 20 ma output. If 10A-5 is set corresponding to the "MARK" condition, 10A-6 is low, current flows from Vcc, out the base of Q3, through R16, and into 10A-6. Q3 is turned on and approximately 20 ma of current will flow from Vcc, out the TTY+ output, in the TTY- output, and through R21 to ground. If the flip-flop is reset, corresponding to the "SPACE" condition, 10A-6 is sufficiently close to Vcc that the voltage at the base of Q3 will not turn on Q3. No current will flow out the TTY+ output in this condition.

When the power on clear signal  $\overline{POC}$  goes low, the flip-flop 10A-5 is set, putting both interface lines into the "MARK" state. This is the appropriate quiescent state for these lines.

### Disk Data Flow

Before discussing the Write Data, Write Mark, Read Data, and Read Mark operations, it is necessary to consider the way that data is organized on the disk.

Each data bit is recorded on a small region of magnetic material that moves under the read/write head. This region is known as a bit cell. Since the disk is rotating under the read/write head at a fixed velocity, the bit cell also corresponds to a region of time.

Bit cells can contain one or two pulses, depending on whether the data recorded is a zero or a one. Bit cells start with a pulse known as the clock pulse. A fixed amount of time later will be a data pulse if the bit cell has a one recorded in it, or no data pulse if a zero is recorded.

A byte of data is recorded on the disk at eight consecutive bit cells. The most significant bit of the byte is the first bit recorded.

For more effective use of the disk surface, information stored on a track of the disk is traditionally divided into fixed size blocks of data called sectors. Sectors consist of a header field identifying track and sector number, a data area, and a checksum. A number of sectors follow each other in series around the circumference of the track. Gaps between the sectors and between the header field and the data area within each sector allow time for the write head to turn on or off so that one of these fields can be written without erasing the following field or sector.

There are two techniques in use to identify where on the track sectors begin. These techniques are referred to as hard sectoring and soft sectoring. Each of them will be discussed below.

Every diskette has a hole punched in it near the hub. This hole, sensed by a phototransistor, identifies the beginning of the track. In the hard sectoring format, a similar hole is punched for each sector, indicating that the sector is about to begin.

Given the small physical size of information on the disk, it is impossible to accurately align a mechanical hole with the data recorded on the disk. The hole therefore only indicates that the track or sector is about to begin. An address mark is recorded just before the data on the disk. After the hole is sensed, it is necessary to wait until this recorded mark is detected, indicating the precise start of a sector. The soft sectored format relies solely on these recorded marks, making sector holes unnecessary.

There are actually four such marks. The Index Mark is used to indicate the beginning of each track. The Sector Mark is used to indicate the beginning of the header field of each sector. The Data Mark is used to indicate the start of the data area of each sector. The Deleted Data Mark can be used in place of the Data Mark to indicate an alternate type of data; in practice it is rarely used. The meaning of this last mark is up to the user.

## Hardware Description

What makes a mark immediately recognizable is that it is missing certain clock pulses. If we encode the clock bits from each bit cell into a byte of hexadecimal in the same manner as the data bits with a present clock pulse being considered a 1 and an absent clock pulse a  $\emptyset$ , we get the following pattern for the four marks:

	Data Bits	Clock Bits
Index Mark	FC	D7
Sector Mark	FE	C7
Data Mark	FB	C7
Deleted Data Mark	F8	C7

The Index Mark and Sector Marks are normally written onto a track only when the disk is initialized. The Deleted Data Mark is not used in standard format disks. The Data Mark is written each time the data area of a sector is updated.

The marks also serve another function: synchronization. Each separately writeable field on the disk must have a gap before and after it in which no meaningful information is recorded to allow time for the write head to turn on before writing and turn off after writing. These gaps are invariably filled with random magnetizations that would make any system lose track of where the byte divisions in the serially recorded bit stream are (to say nothing of which pulses are clock pulses and which pulses are data pulses). By a technique discussed below, the marks, which precede every field written on the disk, can be used to decide unambiguously where the clock pulse of the first bit cell of the first byte of the field begins.

The four disk data transfer operations have certain hardware in common. The basic timing for each bit cell is generated by four D flip-flops in slots 12B and 13B. They are clocked by SYS CLK connected to pins 11 and 3. Arranged as a shift register, the flip-flops cycle through a sequence of eight states before repeating themselves as illustrated in Figure 1.

## Hardware Description

The first state represents the time period in which a clock pulse, if any, will occur. There are then three states which correspond to the interval of three clock periods before the data pulse. Then there is the state corresponding to the data pulse, followed by three more states representing the time from the data pulse to the end of the bit cell.

The counter in slot 13C is used to count the eight bit cells that comprise one byte. It is clocked by  $\overline{C}$  CLK attached to pin 2, so that it changes state at the start of each bit cell. Although it is a divide by sixteen counter, it is used as a divide by eight counter (no connection is made to Q-D). In certain circumstances, it is necessary to set this counter to a certain state for synchronization purposes. This is accomplished by the signal  $\overline{LD}$ , connected to pin 9, which loads the value 1100 binary into the counter on the next clock pulse when it is low. This value comes from the data in pins 6, 5, 4 and 3 which are strapped high or low as appropriate. Pin 6 could actually have been connected to Vcc or ground since  $Q_D$  is not used.

Figure 2 shows the states of this counter, starting with a load operation.

See Timing Diagram under Schematics

Two additional signals are derived from this counter:  $\overline{EOC}$  (End of Character) is generated by gate 13A-6 during the C CLK period of the bit cell where the low order three bits of the counter have the binary value 110;  $\overline{EOW}$  (End of Word) is generated by 13A-8 during the A CLK period of the following bit cell.

Let us now consider the actual data transfer operation.

**WRITE DATA:** When the  $\overline{WRITE DATA}$  line goes low at pin 3C-9, 3C-8 goes high, clocking a  $\emptyset$  into flip-flop 2B-5. 3B-8 goes high, driving  $\overline{READY}$  low. This halts the CPU until proper synchronization is obtained and the byte of data can be accepted from the S-100 data bus into the Disk Jockey.  $\overline{I/O ENBL}$  connected to 3A-15 provides insurance that the board can only halt the CPU when one of its registers is actually selected.

## Hardware Description

Since the bit cell and byte counters will be at an unpredictable point when the Write Data command is issued, it is necessary to wait until they come to the point indicating the start of a byte. This is done by flip-flop 1C-5. When the byte counter reaches the EOC state and A CLK goes high, the 1 present on 2B-6 since the write command was issued (WRITE SYNC) is loaded onto 1C-5, and WRITE GATE goes high. In the next bit cell, EOW goes low, setting 2B-5, so WRITE SYNC goes low. Thus, on the positive edge of A CLK in the next bit cell where EOC is high, WRITE GATE will go low. WRITE GATE will therefore be high for eight consecutive bit cells between two EOCs when a Write Data is being performed.

When WRITE GATE goes high, it drives 10B-1 low. As a result, 11B-9 and 11B-3 stay high. This ensures that the bit cell counter cycles through its normal sequence of states without being set or reset.

Shift register 11C converts the data to be written from its parallel format on the internal data bus to the serial format that must be sent to the disk. It is clocked by the A CLK towards the end of each bit cell to set up the data for the next bit cell. During the first EOC after WRITE SYNC goes high, it is necessary to load the parallel byte of data written by the CPU into the shift register. Pin 5C-8 goes high during this time, driving 11C-9 high. This indicates the load function to the 74LS299 in slot 11-C. Accordingly, when the A CLK goes high, the byte of data from the internal data bus is loaded into the shift register from its bi-directional data pins. These pins are inputs since READ ATTN is low, so 11C-2 is high.

On the next bit cell, the EOW signal becomes active. This resets WRITE SYNC. As a result, 2B-5 goes high, 3B-8 goes low, and PREADY goes high. The CPU is now free to leave the wait state; appropriate since the byte of data to be written to disk has been successfully loaded into shift register 11C.

Output Q-H of 11-C now has the first bit to be written to the disk. Since WRITE SYNC is no longer high, 11C-9 is low, conditioning the 74LS299 for a right shift operation. In subsequent bit cells, EOC will not be high, so 11C-9 will stay low, even if WRITE SYNC comes high again as a result of a subsequent write operation before the termination of the current one. Since 11C is now a shift register, when A CLK goes high at the end of each bit cell of the byte, the shift register shifts one position, presenting the next bit to be written serially to the disk at output Q-H. Gate 12C-8 combines the data bit with clocks to produce the WRITE DATA signal. This signal goes low during the appropriate time of a bit cell if a 1 is to be written to disk; it stays high if a zero is to be written.

Meanwhile, WRITE CLOCK pulses are being generated for each bit cell by 12C-6 at the appropriate time in each bit cell. 12C-2 is always a one since this is not a Write Mark operation and shift register 9B is therefore conditioned to shift in all ones from its serial input 9B-10. A clock pulse is therefore generated for every bit cell.

## Hardware Description

WRITE DATA and WRITE CLOCK are or'ed together by 3C-3 and driven onto the WRITE DISK DATA line to the write head of the disk by the 74368 in slot 3A. The WRITE GATE signal is also driven onto this cable to indicate to the floppy disk drive the time period during which the WRITE DISK DATA is meaningful.

When the last data bit has been shifted out of 11C, the byte has been successfully written to disk. If the CPU has issued another Write Data command while the first byte was being shifted out, this new byte will be loaded into 11C during the EOC period and the process will repeat itself. WRITE GATE will stay high since WRITE SYNC will have been set again. If no further data bytes were written, WRITE SYNC will be low, and WRITE GATE will go low during EOC on the positive edge of A CLK (the end of the last bit cell of the character).

A Write Mark operation is identical to a Write Data operation except that the byte of data is written with certain clock pulses missing. Shift register 9B is responsible for generating the correct pattern of missing clock pulses during the eight bit cells. This register is clocked by A CLK. When WRITE MARK is low, during the EOC period when data is being loaded into the data shift register, 8B-3 goes low and the pattern present at the data inputs to 9B is loaded into it. This pattern is then shifted out during the next eight bit cells analogously to the data bits, with a one providing a clock pulse for the bit cell and a zero providing a missing clock pulse.

The pattern of missing clocks can be a hexadecimal C7 or D7, depending on which mark is to be written. Gates 10B-13 and 5B-6 decode from the data pattern of the mark which clock pattern is appropriate and provide the appropriate level to 9B-3 for parallel loading into the shift register.

**READ MARK:** The Read Mark operation performs two functions: (1) to synchronize the data separator logic (flip-flops 12B and 13B) and the byte counter (13C); and, (2) to read the data pattern of the mark so that software can determine which of the four types of mark it is (see above).

When the READ MARK signal goes low, 5C-6 goes high, clocking a one into 2C-5 (LD and A0 are both high at this time, so 2C-1 is high). The output of OR gate 8B-6 is therefore high, putting the set input 2B-10 into its inactive state.

Meanwhile READ MARK going low causes 3C-6 (READ ATTN) to go high, clocking a 0 into 2B-9. This causes 3B-9 to go low, 3B-8 to go high, so PREADY goes low, putting the CPU into a wait state until the mark is detected.

DISK DATA -- the serial stream of clock and data pulses from the disk enters gate 10B-3 from the disk drive cable. Since this is a read operation, WRITE GATE will be low and DISK DATA will appear at 10B-1. A pulse on this line will be routed to DATA or to reset (RESET) flip-flops 12-B and 13-B

## Hardware Description

depending on whether A CLK is high or low respectively. A CLK therefore defines the time window in which a pulse from DISK DATA is interpreted as a clock or a data pulse. If A CLK is high, the pulse is a data pulse; if it is low, it is a clock pulse. Nominally, data pulses should occur when the flip-flops are all set and clock pulses should occur when the flip-flops are all reset. The pulse from the disk by setting the flip-flops to all zeros will force the flip-flops to this exact synchronization if the pulse occurs any time within the A CLK window.

Register 7A wired to act like a shift register has the function of counting missing clock pulses. It is clocked on the positive edge of A' CLK just after the clock pulse nominally occurs. Since the head is loaded, HEAD is high and ONE comes high. The register is reset by RESET which goes low whenever a clock pulse is detected from the disk. In normal operation, ONE will go high between clock pulses, but be reset by each clock pulse. If a clock pulse is missing, however, the register will not be reset and TWO will come high just after the first missing clock pulse should have occurred. If three missing clock pulses occur in a row, FOUR will come high. As will be discussed in a moment, this condition occurs when an address mark with three missing clock pulses is read. Four missing clock pulses cause OUT OF PHASE to go high. There are only two possible explanations of four missing clock pulses: garbage is being read off the disk or the system has its timing backwards and is interpreting data pulses as clock pulses and vice versa. If the missing "clock pulses" is actually supposed to be a data pulse, the correct response is to reset the system timing and interpret the next pulse as a clock pulse.

As can be seen on the prints, OUT OF PHASE going high forces 10B-4 low and 11B-14 high, routing the next pulse from DISK DATA to the RESET line, i.e., interpreting it as a clock pulse. While this may not help much if garbage is being read off the disk, this technique guarantees that the system will be correctly distinguishing clock and data pulses within several bit cells after the start of a field of data zeros on the disk (since the missing data pulses of the zeros will continually trigger the missing clock detection logic if the system is out of sync). The recording standards for floppy disks require that six bytes of zeros be written before any address mark for this reason. This function is automatically carried out by Disk Jockey firmware when standard subroutines are used to write data sectors. As a result, one can guarantee that clock and data pulses are being correctly interpreted by the start of the address mark. Now it only remains to figure out which bit cell starts a byte of recorded data.

The System Timing Diagram shows the response of the circuitry to the last bit cell of the zero field followed by an address mark. The first two bit cells of the address mark are ones with normal clock pulses. The next three bit cells have missing clock pulses, denoted by dotted lines. As a result, the line FOUR will be high after the third missing clock pulse. On the next positive edge of C CLK, LD will be low, since READ MARK, FOUR and EOW will be high (EOW must be high since A' CLK is low at the positive edge of C CLK). Byte counter 13C will therefore load the value 1100 from its data inputs (13C-6, 5, 4, and 3), establishing synchronization of bit

## Hardware Description

cells, reaching the value 111 at the start of the next byte. As can be seen,  $\overline{EOC}$  will be low during the last bit cell of the byte, and  $\overline{EOW}$  will be low during the first bit cell of the next byte, which is the correct synchronization for these signals.

The  $\overline{LD}$  signal also goes to 5B-1. Since  $A_0$  is high (the address of the Read Mark Register being odd), 2C-1 goes low, 2C-5 goes low, and 8B-4 goes low. This indicates that synchronization of the byte counter has been obtained. As a result, the next time  $\overline{EOW}$  goes low, slightly into the first bit cell of the following byte, 8B-6 goes low, 2B-10 goes low, and 2B-9 goes high. The CPU is therefore allowed to leave the wait state, terminating the READ MARK operation.

Parallel to this activity, the data bits of the mark have been collected. Flip-flop 2C-9 is reset by B CLK during the clock period of each bit cell. It will then be set by DATA if there is a pulse during the data pulse period of the bit cell (corresponding to a recorded 1 bit) or 2C-9 will stay at 0 if there is no DATA pulse corresponding to a recorded 0.

The READ DATA signal generated by this flip-flop goes to the serial input of shift register 11C. 11C is conditioned to act as a shift register since WRITE SYNC is low at 3B-5 so 11C-19 must be low. The data bit is shifted into the register on the positive edge of A CLK, shortly after it is detected.

READ ATTN being high at 11B-13 puts 11C-2 low, enabling the byte stored in the shift register onto the internal data bus. Thus, when the CPU is released from the wait state at the end of a read operation, the last eight data bits collected will be available on the S-100 bus data in lines. The CPU will read this byte before the next positive edge of A CLK reads the next bit into the register.

READ DATA: Read Data is simpler than Read Mark since all counters are already running in sync. The normal state of shift register 11C is to collect the read data coming in from the disk. When the READ DATA line goes low at 3C-5, READ ATTN goes high, enabling the contents of 11C onto the internal data bus and loading flip-flop 2B-9 with a 0, putting the CPU into the wait state. When the byte currently being read has been completely loaded into shift register 11C,  $\overline{EOW}$  goes low. Since the Read Data Register is at an even address,  $A_0$  is low, so 2C-1 is low, 2C-5 is low, and 8B-4 is low. 8B-6 therefore goes low, setting 2B-9 to 1 and the CPU leaves the wait state and reads in the collected byte.

Note that while both READ MARK and READ DATA release the CPU from the wait state slightly into the byte after the one being read, this still leaves adequate time for the CPU to read the current byte and issue a new READ DATA instruction before the new data byte has been assembled from the disk ( $\overline{EOW}$  going low). Since read data is continuously collected in shift register 11C, it is therefore no problem to read a series of bytes from the disk.



Oisk Jockey Firmware

THE DISK JOCKEY SHUGART FIRMWARE

340:000	ORIGIN EQU 340:000Q	340:065	315 305 340		CALL	OREAD	READ IN BOOTSTRAP
343:000	READDATA EQU ORIGIN+300H	070	302 055 340		JNZ	BLOOP	RELOAD ON ERROR
343:001	READMARK EQU READDATA+1	073	311		RET		BRANCH TO BOOTSTRAP
343:002	STATUS EQU READDATA+2	074	166		HLT		TEST INSTRUCTION
343:003	LOADHEAD EQU READDATA+3	075	166		HLT		TEST INSTRUCTION
343:000	WRITEOATA EQU REAOOATA	076	000:002		DS	2	
343:001	WRITEMARK EQU READMARK						
343:002	DISKFUNCT EQU STATUS	100	006 200	INPUT	MVI	B,200Q	INITIALIZE BIT COUNT
343:003	SERIAL EQU LOAOHEAO	102	052 160 342		LHLD	SCON	GET SPEED CONSTANT
342:166	BUFFER EQU ORIGIN+2:166Q	105	021 002 343		LXI	O,STATUS	INITIALIZE STATUS REG A0OR
342:171	SECREG EQU BUFFER+3	110	032	WAIT	LOAX	0	GET STATUS
342:164	OMAAOOR EQU BUFFER-2	111	037		RAR		AND TEST
342:175	OATAMARK EQU BUFFER+7	112	037		RAR		FOR A START
342:375	LASTDATA EQU DATAMARK+200Q	113	332 110 340		JC	WAIT	BIT
342:163	DRIVE EQU BUFFER-3	116	315 205 340		CALL	XLOOP	WAIT 1/2 A BIT TIME
342:160	SCON EQU BUFFER-6	121	032		LOAX	0	GET STATUS
000:010	TZERO EQU 10Q	122	037		RAR		AND TEST
000:024	MOVIN EQU 24Q	123	037		RAR		THAT THE START
000:020	MVOUT EQU 20Q	124	332 100 340		JC	INPUT	BIT IS STILL THERE
000:020	STEP EQU 20Q	127	315 201 340	DATAL	CALL	SOELAY	WAIT ONE BIT TIME
000:010	TCONST EQU 10Q	132	032		LDAX	0	GET
000:143	MSEC EQU 143Q	133	037		RAR		THE
000:043	SETTLE EQU 35	134	037		RAR		SERIAL OATA
		135	170		MOV	A,B	IN B
		136	037		RAR		ROTATE
		137	107		MOV	B,A	SAVE
340:000	DBOOT JMP BOOT	140	322 127 340		JNC	DATAL	TEST FOR BYTE COLLECTED
003	TERMIN JMP INPUT	143	315 201 340		CALL	SDELAY	WAIT ONE BIT TIME
006	TRMOUT JMP OUTPUT	146	315 201 340		CALL	SDELAY	WAIT SECOND BIT TIME
011	TKZERO JMP HOME	151	170		MOV	A,B	CHARACTER TO ACC
014	TRKSET JMP SEEK	152	346 177		ANI	177Q	TRIM PARITY
017	SECTOR JMP SETSEC	154	311		RET		RETURN
022	DMA JMP SETOMA						
025	REAO JMP DREAD						
030	WRITE JMP DWRITE	155	021 003 343	OUTPUT	LXI	O,SERIAL	INITIALIZE SERIAL O/P REG
		160	207		ADD	A	APPEND START BIT
033	BOOT LXI SP,BUFFER	161	016 013		MVI	C,11	TOTAL BIT COUNT
036	LXI H,41Q	163	067		STC		SET REST BITS
041	MVI A,210Q	164	037	OLOOP	RAR		ROTATE
043	LXI B,200Q	165	107		MOV	B,A	SAVE
046	PUSH B	166	237		SBB	A	SEND PRESENT
047	PUSH PSW	167	022		STAX	0	BIT TO OUTPUT
050	PUSH H	170	315 201 340		CALL	SOELAY	DELAY ONE BIT TIME
051	PUSH B	173	170		MOV	A,B	GET NEXT BIT
052	STA STATUS	174	015		DCR	C	DECREMENT BIT COUNT
055	BLOOP CALL HOME	175	302 163 340		JNZ	OLOOP	OUTPUT NEXT BIT
060	MVI C,1	200	311		RET		
062	CALL SETSEC						

Disk Jockey Firmware

341:045	016	001		MVI	C,1	SECTOR NUMBER ONE
047	171		SETSEC	MOV	A,C	GET SECTOR NUMBER
050	346	037		ANI	37Q	TEST IF LE55
052	310			RZ		THAN ONE
053	117			MOV	C,A	SAVE TRIMMED VALUE
054	306	345		AOI	345Q	TEST FOR GREATER
056	237			5BB	A	THAN 26
057	300			RNZ		
060	041	171	342	LXI	H,SECREG	SAVE SECTOR VALUE
063	161			MOV	M,C	& COMPUTE
064	315	116	341	CALL	SETCRC	HEAOER CHECK SUM
067	161			MOV	M,C	SAVE
070	043			INX	H	CHECK SUM
071	160			MOV	M,B	AT END
072	043			INX	H	OF SECTOR
073	066	373		MVI	M,OFBH	HEADER & AL50
075	257			XRA	A	WRITE DATA MARK
076	311			RET		
077	041	000	342	SETDMA	LXI	H,ORIGIN+2:000Q
102	174			MOV	A,H	
103	270			CMP	B	
104	310			RZ		
105	074			INR	A	
106	270			CMP	B	
107	310			RZ		
110	140			MOV	H,B	
111	151			MOV	L,C	
112	042	164	342	5HLD	DMAADDR	
115	311			RET		
116	041	166	342	SETCRC	LXI	H,BUFFER
121	021	172	342		LXI	D,SECREG+1
124	001	377	377	CRECH	LXI	B,-1
127	325			PUSH	O	
130	176			MOV	A,M	
131	251			XRA	C	
132	127			MOV	D,A	
133	017			RRC		
134	017			RRC		
135	017			RRC		
136	017			RRC		
137	346	017		ANI	OFH	
141	252			XRA	O	
142	137			MOV	E,A	
143	017			RRC		
144	017			RRC		
145	017			RRC		
146	127			MOV	D,A	
147	346	037		ANI	1FH	
151	250			XRA	B	

Disk Jockey Firmware

341:152	117			MOV	C,A	
153	172			MOV	A,D	
154	346	340		ANI	OEOH	
156	253			XRA	E	
157	107			MOV	B,A	
160	172			MOV	A,D	
161	017			RRC		
162	346	360		ANI	OFOH	
164	251			XRA	C	
165	117			MOV	C,A	
166	043			INX	H	
167	321			POP	D	
170	172			MOV	A,D	
171	274			CMP	H	
172	330			RC		
173	302	127	341	JNZ	CRECH+3	
176	173			MOV	A,E	
177	275			CMP	L	
200	330			RC		
201	303	127	341	JMP	CRECH+3	
204	315	355	341	SEEK	CALL	LDHEAD
207	300				RNZ	SELECT DRIVE, LOAD HEAD
210	171				MOV	& TEST READY
211	346	177			ANI	GET TRACK VALUE
213	117				MOV	TRIM &
214	306	263			ADI	SAVE
216	237				5BB	TEST FOR
217	300				RNZ	GREATER THAN 76
220	041	167	342		LXI	H,BUFFER+1
223	176			SLOOP	MOV	A,M
224	271				CMP	GET CURRENT TRACK
225	312	337	341		JZ	COMPARE WITH NEW TRACK
230	332	241	341		JC	HEAD SETTLE DELAY
233	006	020			MVI	NEW TRK GREATER THAN OLD TRK
235	065				DCR	NEW TRK LESS THAN OLD TRK
236	303	244	341		JMP	
241	006	024		MOVEIN	MVI	B,MOVEIN+3
243	064				INR	M
244	315	322	341		CALL	TDELAY
247	303	223	341		JMP	5LOOP
252	315	355	341	HOME	CALL	LDHEAD
255	300				RNZ	START DRIVE, LOAD HEAD
256	006	024		HLOOP	MVI	& TEST FOR READY
260	315	322	341		CALL	B,MOVEIN
263	032				CALL	TDELAY
264	346	010			LOAX	D
266	302	256	341		ANI	TZERO
271	006	020		5TEPO	JNZ	HLOOP
273	315	322	341		MVI	B,MVOUT
					CALL	TDELAY
						STEP TOWARD
						TRACK ZERO

*Handwritten notes:*  
 MOVE A, 342  
 CALL 177Q  
 COMP B  
 MOV H, B  
 MOV L, C  
 DMAADDR  
 19  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50  
 51  
 52  
 53  
 54  
 55  
 56  
 57  
 58  
 59  
 60  
 61  
 62  
 63  
 64  
 65  
 66  
 67  
 68  
 69  
 70  
 71  
 72  
 73  
 74  
 75  
 76  
 77  
 78  
 79  
 80  
 81  
 82  
 83  
 84  
 85  
 86  
 87  
 88  
 89  
 90  
 91  
 92  
 93  
 94  
 95  
 96  
 97  
 98  
 99  
 100

Disk Jockey Firmware

340:201	052	16D	342	SDELAY	LHLD	SCON	INITIALIZE
204	051				DAD	H	SERIAL DELAY CONSTANT
205	053			XLOOP	DCX	H	DECREMENT COUNT
206	174				MOV	A,H	TEST FOR
207	265				ORA	L	COUNT
210	302	205	340		JNZ	XLOOP	EQUAL TO ZERO
213	311				RET		
214	315	355	341	DWRITE	CALL	LDHEAD	LOAD THE HEAD
217	032				LDAX	D	GET STATUS
220	346	0D1			ANI	1	TEST FOR
222	300				RNZ		WRITE PROTECT
223	052	164	342		LHLD	DMAADDR	GET DMA ADDRESS
226	021	175	342		LXI	D,DATAMARK	
231	325				PUSH	D	SAVE POINTER
232	315	363	340		CALL	XFER	LOAD DISK DATA BUFFER
235	341				PDP	H	RECOVER POINTER
236	315	124	341		CALL	CRECH	CALCULATE CRC WORD
241	161				MOV	M,C	STORE
242	043				INX	H	-CRC
243	160				MOV	M,B	-WORD
244	016	0D7			MVI	C,7	NUMBER OF HEADER BYTES+1
246	315	377	340		CALL	RSECT	FIND RIGHT SECTOR
251	300				RNZ		RETURN IF DISK ERROR
252	0D6	012			MVI	B,10	DELAY GAP
254	032			FLOOP	LDAX	D	DUMMY READ
255	0D5				DCR	B	WAIT FOR 1D BYTES
256	302	254	340		JNZ	FLDDP	BEFORE WRITING
261	257				XRA	A	PREPARE TO WRITE ZEROS
262	022			ZLDDP	STAX	O	WRITE
263	015				DCR	C	SIX BYTES
264	302	262	340		JNZ	ZLDDP	OF ZEROS
267	023				INX	D	
270	176				MOV	A,M	GET THE
271	022				STAX	D	DATA MARK & WRITE
272	033				DCX	D	ADJUST BACK TO DATA
273	043				INX	H	ADJUST POINTER
274	176			WLDDP	MOV	A,M	GET DATA BYTE
275	022				STAX	D	WRITE IT ON THE DISK
276	054				INR	L	MOVE POINTER AHEAD
277	302	274	340		JNZ	WLOOP	& TEST FOR LAST BYTE
302	257				XRA	A	PREPARE TO WRITE A ZERO
303	022				STAX	O	WRITE FINAL ZERO
304	311				RET		
305	315	377	340	DREAD	CALL	RSECT	FIND CORRECT SECTOR
310	300				RNZ		RETURN IF DISK ERROR
311	006	016			MVI	B,14	GAP WIDTH
313	032			RWAIT	LDAX	D	READ PAST

Disk Jockey Firmware

340:314	0D5				DCR	B	CRAP IN
315	302	313	340		JNZ	RWAIT	THE GAP
320	072	001	343		LDA	READMARK	WAIT FOR A MARK
323	276				CMP	M	LOOK FOR ANOTHER
324	302	305	340		JNZ	DREAD	SECTOR IF NO MARK
327	054				INR	L	ADVANCE POINTER
330	032			RLOOP	LOAX	D	GET DISK DATA
331	167				MOV	M,A	STORE IN BUFFER
332	054				INR	L	ADVANCE POINTER
333	302	330	340		JNZ	RLOOP	& TEST IF DONE
336	055				DCR	L	BACK UP POINTER
337	353				XCHG		SAVE IN D-E PAIR
340	041	175	342		LXI	H,0ATAMARK	
343	315	124	341		CALL	CRECH	CALCULATE CRC
346	171				MOV	A,C	& TEST
347	26D				ORA	B	FOR ZERO
350	076	001			MVI	A,1	
352	300				RNZ		RETURN IF CRC ERROR
353	021	176	342		LXI	D,DATAMARK+1	
356	052	164	342		LHLO	DMAADOR	
361	353				XCHG		GET READY FOR DMA XFER
362	033				DCX	D	
363	0D6	2D0		XFER	MVI	B,2D0Q	
365	176				MOV	A,M	GET DISK DATA
366	023				INX	O	
367	022				STAX	D	STORE IN MEMORY
370	043				INX	H	
371	005				DCR	B	DECREMENT BYTE COUNT
372	302	365	340		JNZ	XFER+2	& TEST FOR DONE
375	170				MOV	A,B	
376	311				RET		
341:002	3D0				RSECT	CALL	LDHEAD
003	041	166	342		RNZ		LOAD HEAD &
006	006	0D7			LXI	H,BUFFER	TEST FOR READY
010	033				MVI	B,7	SECTOR HEADER MARK
011	033				DCX	D	SECTOR HEADER COUNT
012	032				DCX	D	ADJUST TO READ
013	247			ZWAIT	LDAX	D	DISK DATA
014	302	012	341		ANA	A	READ DISK DATA
017	072	001	343		JNZ	ZWAIT	SET FLAGS
022	276				LDA	READMARK	TEST FOR ZERO
023	302	377	340		CMP	M	READ A MARK
026	043				JNZ	RSECT	COMPARE WITH HEADER
027	0D5				INX	H	READ NEXT HEADER IF ERROR
030	310				DCR	B	ADVANCE BUFFER POINTER
031	032			TSLOOP	RZ		DECREMENT HEADER COUNT
032	276				LDAX	D	RETURN IF DONE
033	312	026	341		CMP	M	READ DISK DATA
036	076	005			JZ	TSLOOP	COMPARE WITH MEMORY
040	22D				MVI	A,5	READ MORE IF NO ERROR
041	37D				SUB	B	TEST FOR
042	3D3	377	340		RM		WRONG TRACK
					JMP	RSECT	NUMBER
							NOT A TRACK ERROR

Disk Jockey Firmware

E020	CO AA E1	BLOOP	CALL	HOME
E030	DE D1		MVI	C,1
E032	CD 27 E1		CALL	SETSEC
E035	CD C5 ED		CALL	DREAD
E038	C2 20 E0		JNZ	BLOOP
E038	C9		RET	
E03C	76		HLT	
E03D	76		HLT	
E03E	D002		DS	2
E040	D6 80	INPUT	MV1	B,2DDQ
E042	2A 70 E2		LHLO	SCON
E045	11 02 E3		LXI	D,STATUS
E048	1A	WAIT	LOAX	D
E049	1F		RAR	
E04A	1F		RAR	
E04B	0A 4B E0		JC	WAIT
E04E	CD 85 E0		CALL	XLOOP
E051	1A		LDAX	0
E052	1F		RAR	
E053	1F		RAR	
E054	0A 40 E0		JC	INPUT
E057	CO 81 E0	DATAL	CALL	SOELAY
E05A	1A		LDAX	0
E05B	1F		RAR	
E05C	1F		RAR	
E05D	78		MOV	A,B
E05E	1F		RAR	
E05F	47		MOV	B,A
E06D	D2 57 E0		JNC	DATAL
E063	CD 81 E0		CALL	SDELAY
E066	CO 81 ED		CALL	SDELAY
E069	78		MOV	A,B
E06A	E6 7F		ANI	177Q
E06C	C9		RET	
E060	11 03 E3	OUTPUT	LX1	D,SERIAL
E07D	87		ADD	A
E071	0E 0B		MVI	C,11
E073	37	OLOOP	STC	
E074	1F		RAR	
E075	47		MOV	B,A
E076	9F		SBB	A
E077	12		D	STAX
E07B	CD 81 ED		CALL	SDELAY
E07B	78		MOV	A,B
E07C	DD		DCR	C
E07D	C2 73 E0		JNZ	OLOOP
E080	C9		RET	

09

Disk Jockey Firmware

E0B1	2A 7D E2	SDELAY	LHLD	SCON
E0B4	29		DAO	H
E0B5	28	XLOOP	DCX	H
E0B6	7C		MOV	A,H
E0B7	85		ORA	L
E0B8	C2 85 E0		JNZ	XLOOP
E0BB	C9		RET	
E0BC	CO E0 E1	DWRITE	CALL	LDHEAO
E0BF	1A		LDAX	0
E090	E6 01		ANI	1
E092	CO		RNZ	
E093	2A 74 E2		LHLD	DMAADOR
E096	11 7D E2		LXI	0,DATAMARK
E099	05		PUSH	D
E09A	CO F3 ED		CALL	XFER
E09D	E1		POP	H
E09E	CO 54 E1		CALL	CRECH
EDA1	71		MOV	M,C
EDA2	23		INX	H
EOA3	70		MOV	M,B
EOA4	DE D7		MVI	C,7
EOA6	CD FF E0		CALL	RSECT
EOA9	CO		RNZ	
EOAA	06 0A		MVI	B,10
EOAC	1A	FLOOP	LDAX	0
EOAD	05		OCR	B
EOAE	C2 AC E0		JNZ	FLOOP
EOB1	AF		XRA	A
EOB2	12	ZLOOP	STAX	0
EOB3	00		OCR	C
EOB4	C2 B2 E0		JNZ	ZLOOP
EOB7	13		INX	0
EOBB	7E		MOV	A,M
EOB9	12		STAX	D
EOBA	1B		OCX	0
EOBB	23		INX	H
EOBC	7E	WLOOP	MOV	A,M
EOBD	12		STAX	0
EOBE	2C		1NR	L
EOBF	C2 BC E0		JNZ	WLOOP
EOC2	AF		XRA	A
EOC3	12		STAX	0
EOC4	C9		RET	
EOC5	CO FF E0	DREAD	CALL	RSECT
EOC8	CO		RNZ	
EOC9	D6 0E		MVI	B,14
EOCB	1A	RWAIT	LOAX	0
EOCC	05		OCR	B
EOCD	C2 CB E0		JNZ	RWAIT
EOD0	3A 01 E3		LDA	READMARK
EOD3	BE		CMP	M
EOD4	C2 C5 E0		JNZ	OREAO

Disk Jockey Firmware

```

341:276 032          LDAX 0          GET DISK STATUS &
277 346 010        ANI  TZERO      TEST FOR TRACK ZERO
301 312 271 341    JZ   STEPO
304 257           XRA   A
305 041 166 342    LXI  H,BUFFER  SET UP DATA
310 066 376        MVI  M,OFEH  BUFFER
312 054           INR  L          FOR TRACK
313 312 045 341    JZ   SETSEC-2   ZERO &
316 167           MOV  M,A      SECTOR ZERO
317 303 312 341    JMP  ILOOP

322 072 163 342    TOELAY LOA  DRIVE    GET ACTIVE DRIVE NUMBER
325 365           PUSH PSW
326 346 353        ANI  353Q
330 250           XRA  B          MERGE DIRECTION & STEP
331 022           STAX 0          SENO TO ORIVE
332 356 D20        XRI  STEP     FINISH STEP PULSE
334 022           STAX  D        SENO TO ORIVE
335 361           POP  PSW
336 022           STAX  0
337 006 010        MVI  B,TCONST  HEAO MOTION CONSTANT
341 076 143        OELAY MVI  A,MSEC    OELAY
343 177           MOV  A,A
344 075           OCR  A          ONE
345 302 343 341    JNZ  DELAY+2   MILLESECOND
350 005           DCR  B          TEST FOR OELAY
351 302 341 341    JNZ  OELAY     ONE
354 311           RET

355 D21 002 343    LDHEAO LXI  O,STATUS  DISK STATUS MEM LOC
360 032           LOAX 0          GET STATUS BYTE
361 346 0D4        ANI  4          STRIP OFF HEAD LOAO BIT
363 072 003 343    LOA  LOADHEAD  LOAO HEAO & SELECT DRIVE
366 006 043        MVI  B,SETTLE  HEAO LOAO SETTLE TIME
370 314 341 341    CZ   OELAY     CONDITIONALLY WAIT FOR SETTLE
373 032           LOAX 0          GET STATUS
374 027           RAL          SET ACC TO 377Q
375 237           SBB  A          OR ZERO THE ACC FOR
376 074           INR  A          REAOY FLAG TO SYSTEM
377 311           RET
    
```

59

Disk Jockey Shugart Firmware

```

E000          ORIGIN EQU 340:000Q
E300          READDATA EQU DRIGIN+300H
E301          READMARK EQU READATA+1
E302          STATUS EQU REAOOATA+2
E303          LOADHEAD EQU REAOOATA+3

E300          WRITEDATA EQU READDATA
E301          WRITEMARK EQU REAOOATA
E302          OISKFNCT EQU STATUS
E303          SERIAL EQU LOADHEAO
E276          BUFFER EQU ORIGIN+2:166Q
E279          SECREG EQU BUFFER+3
E274          OMAAOOR EQU BUFFER-2
E270          OATAMARK EQU BUFFER+7
E2F0          LASTOATA EQU OATAMARK+200Q
E273          DRIVE EQU BUFFER-3
E270          SCON EQU BUFFER-6

000B          TZERO EQU 10Q
0014          MOVIN EQU 24Q
0010          MVOUT EQU 20Q
0010          STEP EQU 20Q
0008          TCONST EQU 10Q
0063          MSEC EQU 143Q
0023          SETTLE EQU 35

E000 C3 1B E0  OBOOT JMP BOOT
E003 C3 40 E0  TERMIN JMP INPUT
E006 C3 60 E0  TRMOUT JMP OUTPUT
E009 C3 AA E1  TKZERO JMP HOME
E00C C3 B4 E1  TRKSET JMP SEEK
E00F C3 27 E1  SECTOR JMP SETSEC
E012 C3 3F E1  DMA JMP SETOMA
E015 C3 C5 E0  REAO JMP OREAO
E01B C3 BC E0  WRITE JMP DWRITE

E01B 31 76 E2  BOOT LXI SP,BUFFER
E01E 21 21 00  LXI H,41Q
E021 3E BB     MVI A,21DQ
E023 01 80 00  LXI B,200Q
E026 C5       PUSH B
E027 F5       PUSH PSW
E02B E5       PUSH H
E029 C5       PUSH B
E02A 32 02 E3  STA STATUS
    
```

Disk Jockey Firmware

E16A	4F		MOV	C,A
E168	7A		MOV	A,D
E16C	E6 E0		ANI	OEOH
E16E	AB		XRA	E
E16F	47		MOV	B,A
E17D	7A		MOV	A,D
E171	DF		RRC	
E172	E6 FD		ANI	DFDH
E174	A9		XRA	C
E175	4F		MOV	C,A
E176	23		INX	H
E177	D1		PDP	D
E178	7A		MOV	A,D
E179	BC		CMP	H
E17A	DB		RC	
E17B	C2 57 E1		JNZ	CRECH+3
E17E	7B		MOV	A,E
E17F	BD		CMP	L
E180	D8		RC	
E181	C3 57 E1		JMP	CRECH+3
E184	CD ED E1	SEEK	CALL	LDHEAD
E187	CD		RNZ	
E188	79		MOV	A,C
E189	E6 7F		ANI	177Q
E18B	4F		MOV	C,A
E18C	C6 B3		ADI	179
E18E	9F		SBB	A
E18F	CD		RNZ	
E19D	21 77 E2		LXI	H,BUFFER+1
E193	7E	SLDDP	MOV	A,M
E194	B9		CMP	C
E195	CA DF E1		JZ	DELAY-2
E198	DA A1 E1		JC	MOVEIN
E198	D6 10		MVI	B,MVOUT
E19D	35		DCR	M
E19E	C3 A4 E1		JMP	MOVEIN+3
E1A1	D6 14	MOVEIN	MVI	B,MOVIN
E1A3	34		INR	M
E1A4	CD D2 E1		CALL	TDELAY
E1A7	C3 93 E1		JMP	SLDDP
E1AA	CD ED E1	HDME	CALL	LDHEAD
E1AD	CD		RNZ	
E1AE	D6 14	HLOOP	MVI	B,MOVIN
E1BD	CD D2 E1		CALL	TDELAY
E1B3	1A		LDAX	D
E1B4	E6 08		ANI	TZERD
E1B6	C2 AE E1		JNZ	HLODP
E1B9	D6 10	STEPD	MVI	B,MVDUT

Disk Jockey Firmware

E18B	CD D2 E1		CALL	TDELAY
E18E	1A		LDAX	D
E18F	E6 D8		ANI	TZERO
E1C1	CA 89 E1		JZ	STEPO
E1C4	AF		XRA	A
E1C5	21 76 E2		LXI	H,BUFFER
E1C8	36 FE		MVI	M,OFEH
E1CA	2C	ILOOP	INR	L
E1CB	CA 25 E1		JZ	SETSEC-2
E1CE	77		MOV	M,A
E1CF	C3 CA E1		JMP	ILOOP
E1D2	3A 73 E2	TDELAY	LDA	DRIVE
E1D5	F5		PUSH	PSW
E1D6	E6 EB		ANI	353Q
E1DB	AB		XRA	B
E1D9	12		STAX	D
E1DA	EE 1D		XRI	STEP
E1DC	12		STAX	D
E1DD	F1		PDP	PSW
E1DE	12		STAX	D
E1DF	06 D8		MVI	B,TCONST
E1E1	3E 63	DELAY	MVI	A,MSEC
E1E3	7F		MOV	A,A
E1E4	3D		DCR	A
E1E5	C2 E3 E1		JNZ	DELAY+2
E1E8	D5		DCR	B
E1E9	C2 E1 E1		JNZ	DELAY
E1EC	C9		RET	
E1ED	11 D2 E3	LDHEAD	LXI	D,STATUS
E1FD	1A		LDAX	D
E1F1	E6 D4		ANI	4
E1F3	3A D3 E3		LDA	LDADHEAD
E1F6	06 23		MVI	B,SETTLE
E1F8	CC E1 E1		CZ	DELAY
E1F8	1A		LDAX	D
E1FC	17		RAL	
E1FD	9F		SBB	A
E1FE	3C		INR	A
E1FF	C9		RET	

## Disk Jockey Firmware

E0D7	2C		INR	L
E0DB	1A	RLOOP	LOAX	O
E0D9	77		MOV	M,A
EDDA	2C		INR	L
E0DB	C2 0B E0		JNZ	RLOOP
E0DE	2D		OCR	L
E0DF	EB		XCHG	
E0E0	21 70 E2		LXI	H,0ATAMARK
E0E3	C0 54 E1		CALL	CRECH
E0E6	79		MOV	A,C
E0E7	80		ORA	B
E0E8	3E 01		MVI	A,1
E0EA	C0		RNZ	
E0EB	11 7E E2		LXI	D,DATAMARK+1
E0EE	2A 74 E2		LHLO	DMAA00R
E0F1	EB		XCHG	
E0F2	1B		DCX	D
EDF3	06 80	XFER	MVI	B,200Q
E0F5	7E		MOV	A,M
EDF6	13		INX	O
E0F7	12		STAX	O
E0FB	23		INX	H
E0F9	05		DCR	B
E0FA	C2 F5 E0		JNZ	XFER+2
EDFD	7B		MOV	A,B
EDFE	C9		RET	
EDFF	CD ED E1	RSECT	CALL	LDHEAD
E102	C0		RNZ	
E103	21 76 E2		LXI	H,BUFFER
E106	D6 07		MVI	B,7
E108	1B		DCX	D
E109	1B		DCX	D
E10A	1A	ZWAIT	LOAX	D
E10B	A7		ANA	A
E10C	C2 0A E1		JNZ	ZWAIT
E10F	3A D1 E3		LDA	READMARK
E112	BE		CMP	M
E113	C2 FF E0		JNZ	RSECT
E116	23	TSLOOP	INX	H
E117	05		DCR	B
E118	CB		RZ	
E119	1A		LOAX	O
E11A	BE		CMP	M
E11B	CA 16 E1		JZ	TSLOOP
E11E	3E 05		MVI	A,5
E120	9D		SUB	B
E121	FB		RM	
E122	C3 FF E0		JMP	RSECT

## Disk Jockey Firmware

E125	0E 01		MVI	C,1
E127	79	SETSEC	MOV	A,C
E128	E6 1F		ANI	37Q
E12A	CB		RZ	
E12B	4F		MOV	C,A
E12C	C6 E5		ADI	345Q
E12E	9F		SBB	A
E12F	C0		RNZ	
E130	21 79 E2		LXI	H,5CREG
E133	71		MOV	M,C
E134	C0 4E E1		CALL	5ETCRC
E137	71		MOV	M,C
E13B	23		INX	H
E139	70		MOV	M,B
E13A	23		INX	H
E13B	36 FB		MVI	M,DFBH
E130	AF		XRA	A
E13E	C9		RET	
E13F	21 00 E2	SETDMA	LXI	H,ORIGIN+2:00Q
E142	7C		MOV	A,H
E143	BB		CMP	B
E144	CB		RZ	
E145	3C		INR	A
E146	BB		CMP	B
E147	CB		RZ	
E148	6D		MOV	H,B
E149	69		MOV	L,C
E14A	22 74 E2		SHLD	OMAADDR
E140	C9		RET	
E14E	21 76 E2	SETCRC	LXI	H,BUFFER
E151	11 7A E2		LXI	D,5CREG+1
E154	D1 FF FF	CRECH	LXI	B,-1
E157	D5		PUSH	D
E158	7E		MOV	A,M
E159	A9		XRA	C
E15A	57		MOV	D,A
E15B	0F		RRC	
E15C	0F		RRC	
E15D	0F		RRC	
E15E	0F		RRC	
E15F	E6 DF		ANI	DFH
E161	AA		XRA	O
E162	5F		MOV	E,A
E163	0F		RRC	
E164	DF		RRC	
E165	0F		RRC	
E166	57		MOV	D,A
E167	E6 1F		ANI	1FH
E169	A8		XRA	B

## Acknowledgment

Like many other Thinker Toy products designed by Morrow, the Disk Jockey I controller utilizes the power of the CPU to accomplish many of its I/O tasks. However, unlike other Thinker Toy products, the key idea which makes the interface work is not the "brain child" of George Morrow. The Disk Jockey controller uses the CPU's "READY" line in a novel fashion. This unusual use of READY first appeared in an article by Eugene Fisher in the November 8, 1975 issue of Electronics Design Magazine. Mr. Fisher's penetrating insight into the effective use of the microcomputer to control a floppy disk has served as an inspiration in the design of the Disk Jockey controller.

## Warranty

Parts are warranted to be free from defects in material and workmanship. Parts for the Disk Jockey I purchased in kit form is warranted for ninety days from invoice/purchase date. The Disk Jockey I purchased as an assembled unit or as part of the DISCUS I system is warranted for six months from invoice/purchase date. Any board purchased in kit form which is returned for testing/repair is subject to a fee of up to \$35.00. Any out-of-warranty repair of up to \$35.00 will be made without prior approval of customer.

Parts and labor warranty for the disk drive is for forty-five days from invoice/purchase date. For a period of up to one year, there is a flat \$55.00 labor charge for warranty parts replacement. After one year, charges will be made for parts and labor.

Warranty is void if in the opinion of Morrow/Thinker Toys the unit has been subject to abuse, misuse, improper assembly, or if directions have not been followed in assembly.

A COPY OF THE INVOICE OR PROOF OF PURCHASE IS REQUIRED FOR IN-WARRANTY SERVICE. A description of the problem must accompany any item returned for repair. Shipments must be made to Thinker Toys prepaid. Morrow/Thinker Toys is not responsible for any consequential damage.

The foregoing warranty is in lieu of all other warranties expressed or implied and in any event is limited to product repair or replacement.



Morrow Designs, Inc.

  
**Thinker**  
**Toys™**

5221 Central Avenue, Richmond, CA 94804 (415) 524-2101

LIMITED WARRANTY

Morrow Designs Inc. warrants its products to be free from defects in workmanship and material for the period indicated. This warranty is limited to the repair or replacement of parts only and liability is limited to the purchase price of the product. The warranty is void if, in the sole opinion of Morrow Designs Inc., the product has been subject to abuse, misuse, unauthorized modification, improper assembly, non-conformance to assembly directions, or if the unit is used in any other manner than intended.

KITS - Parts, including the printed circuit boards, purchased in kit form are warranted for a period of ninety (90) days from the invoice/purchase date. If a board, which was purchased in kit form, is returned for testing or repair, a minimum service charge of \$35. will be assessed.

ASSEMBLED BOARDS - Parts, including the printed circuit boards, purchased as factory assemblies, are warranted for a period of six (6) months from the invoice/purchase date. Out-of-Warranty boards returned for testing or repair will be assessed a minimum of \$35. service charge. If the charge to repair will exceed \$35., the customer will be notified prior to the actual repair.

ELECTROMECHANICAL PERIPHERALS - Peripheral equipment, such as floppy disk drives, hard disk drives, etc., not manufactured by Morrow Designs Inc. have warranties which vary according to the manufacturer. In most cases, Morrow Designs Inc. provides a warranty equal to or greater than the original manufacturer. Please contact the factory for individual warranty information. Warranty information for each device is included with the equipment when it is shipped.

RETURN PROCEDURE - A COPY OF THE INVOICE OR PROOF OF ORIGINAL PURCHASE IS REQUIRED AND MUST ACCOMPANY THE ITEM FOR IN-WARRANTY SERVICE. Items returned without proof of original purchase will be sent back, shipping charges collect. A description of the problem must accompany the returned item. Shipment must be made prepaid to Morrow Designs, Inc. Repaired items will be shipped via U.P.S. surface. Shipment by air requires payment of the additional charges. Morrow Designs Inc. is not responsible for any consequential damages or for damage incurred in transit.

The foregoing warranty is in lieu of all other warranties either expressed or implied and, in any event, is limited to product repair or replacement.

Effective February 1, 1980

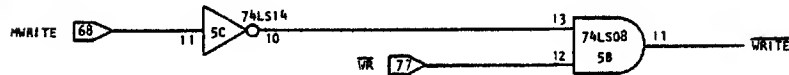
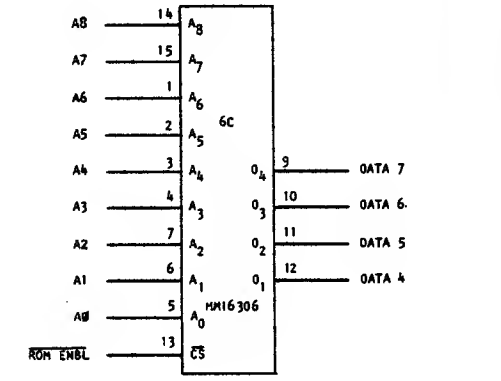
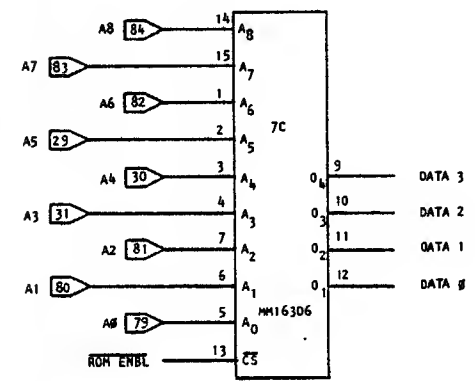
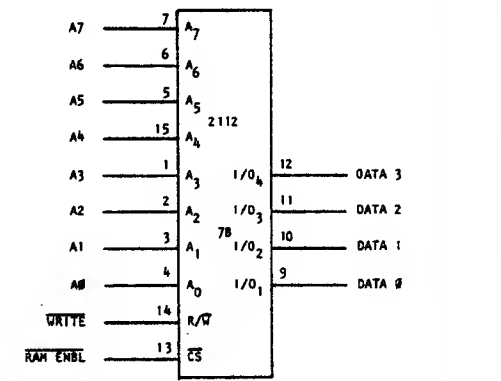
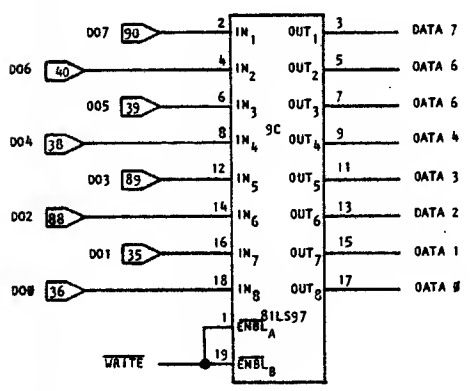
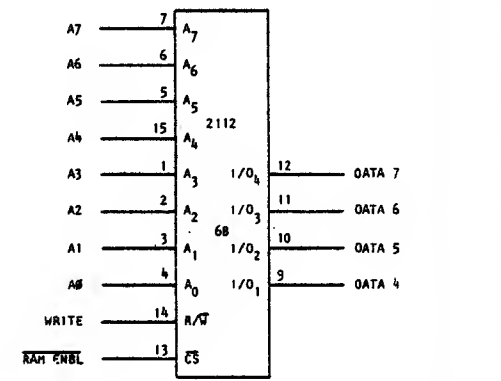
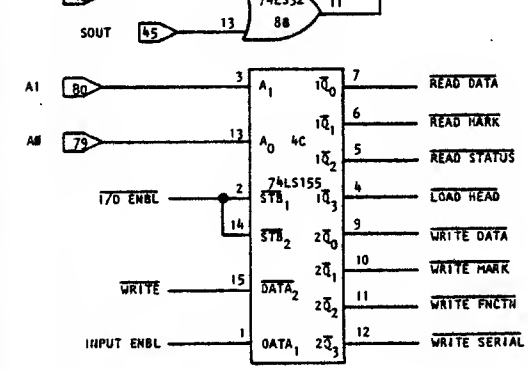
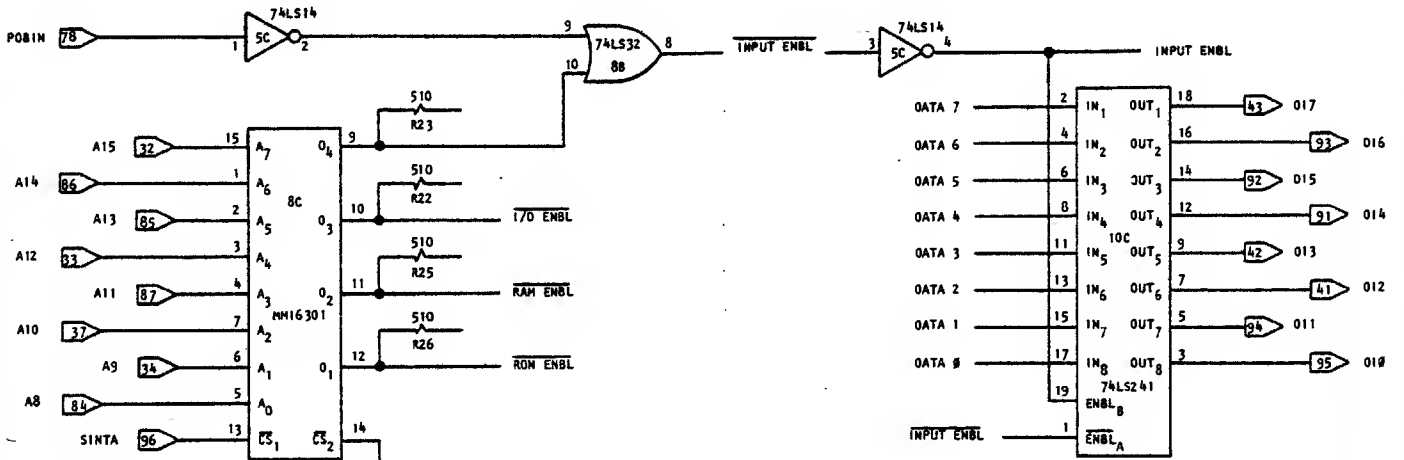
Specifications, terms, and pricing are subject to change without notice.

LIMITED WARRANTY

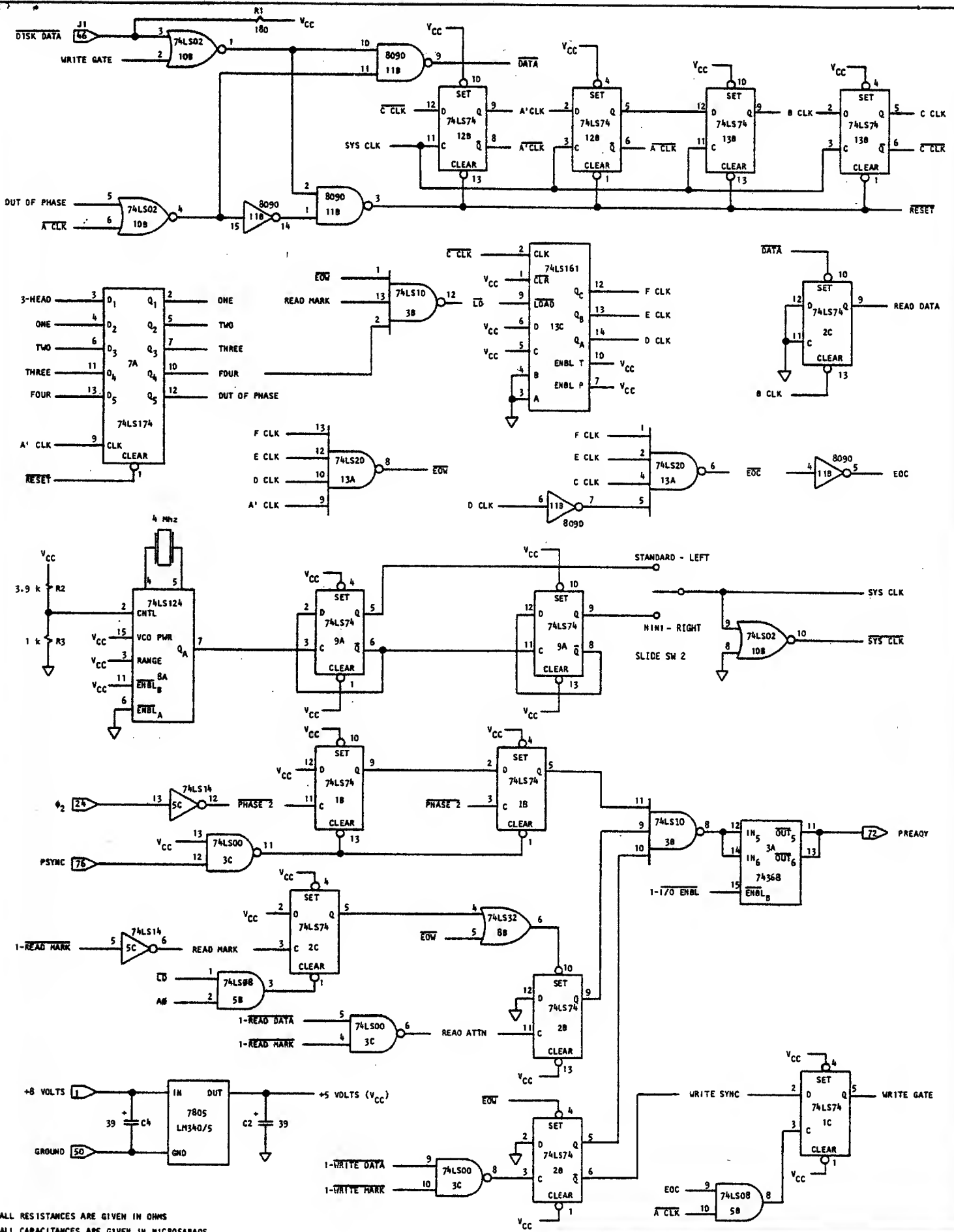
DISCUS 1 and DISCUS 2D Systems

This addendum to Morrow Designs Inc. Limited Warranty applies to the Shugart Associates Model 800/801 Floppy Disk Drives as used in the DISCUS 1 and 2D Disk systems.

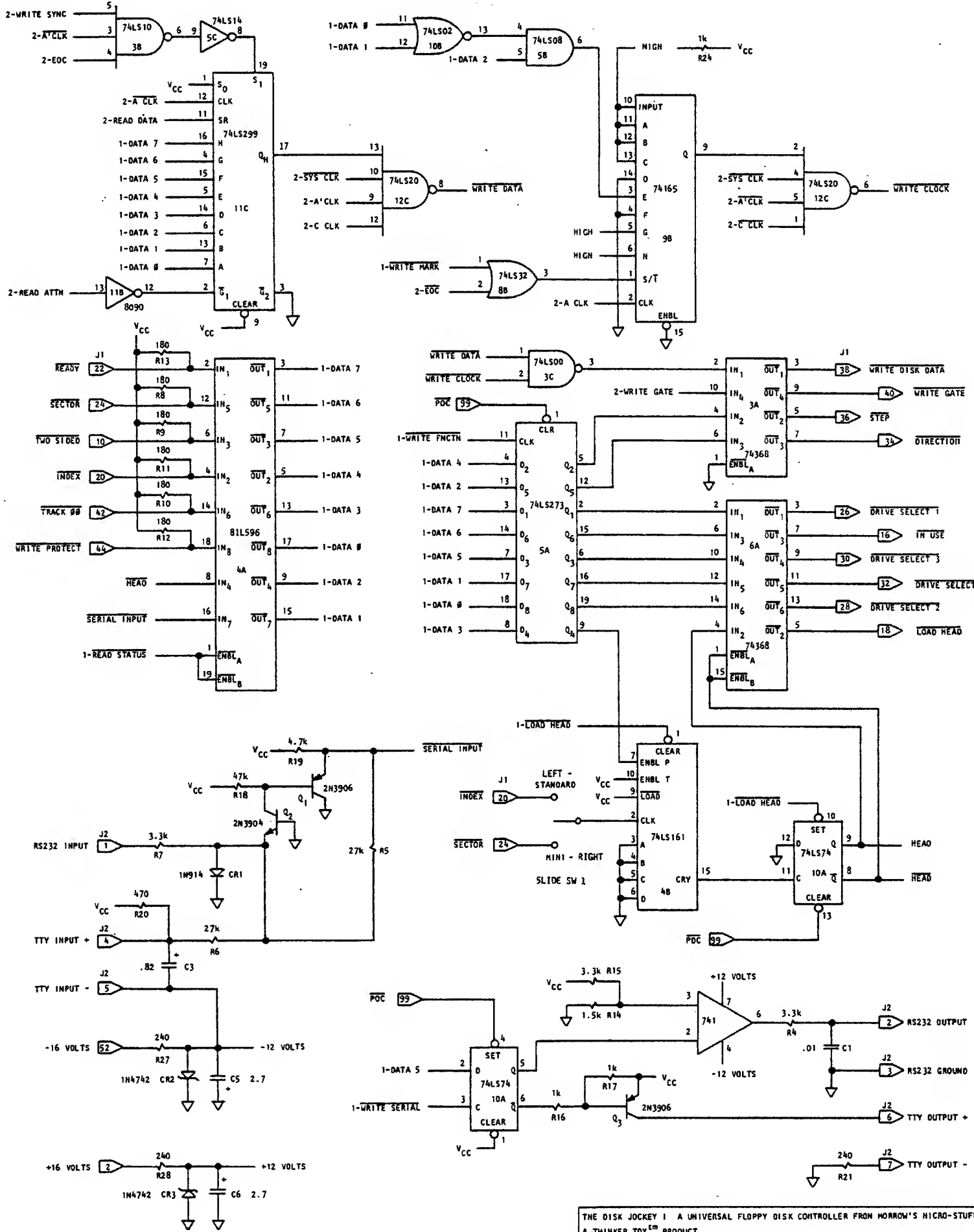
Parts and labor for a floppy disk drive purchased from Morrow Designs Inc. are warranted for a period of forty-five (45) days from the invoice/purchase date. For a period of one (1) year from the invoice/purchase date, parts are warranted. A fixed fee of \$55. will be charged for labor. After one (1) year current rates for parts and labor will be charged.



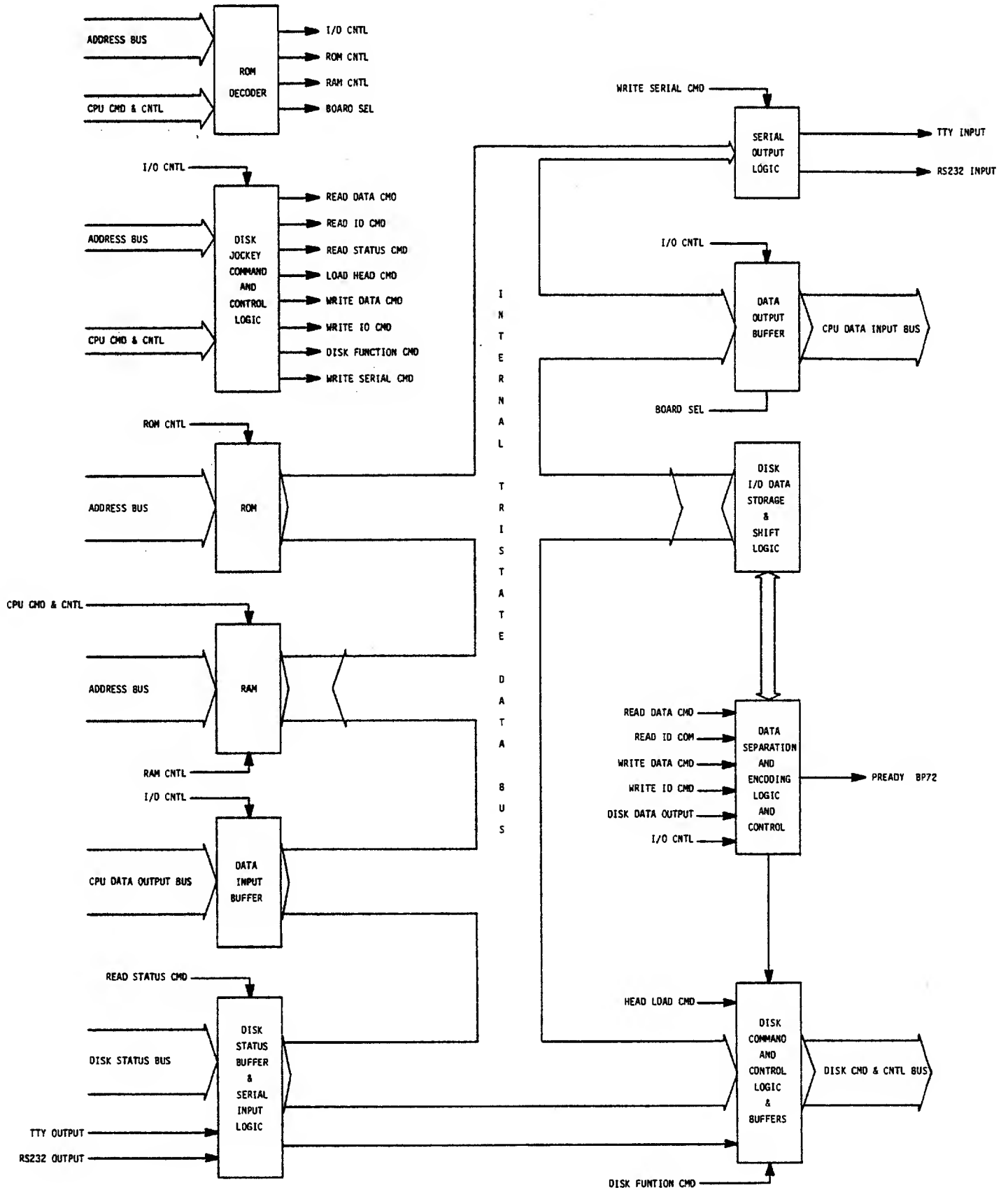
THE DISK JOCKEY I A UNIVERSAL FLOPPY DISK CONTROLLER FROM HORROW'S MICRO-STUFF  
 A THINKER TOY™ PRODUCT  
 DATA BUFFERS, DEVICE SELECTION, RAM, AND ROM  
 PAGE 1 OF 3



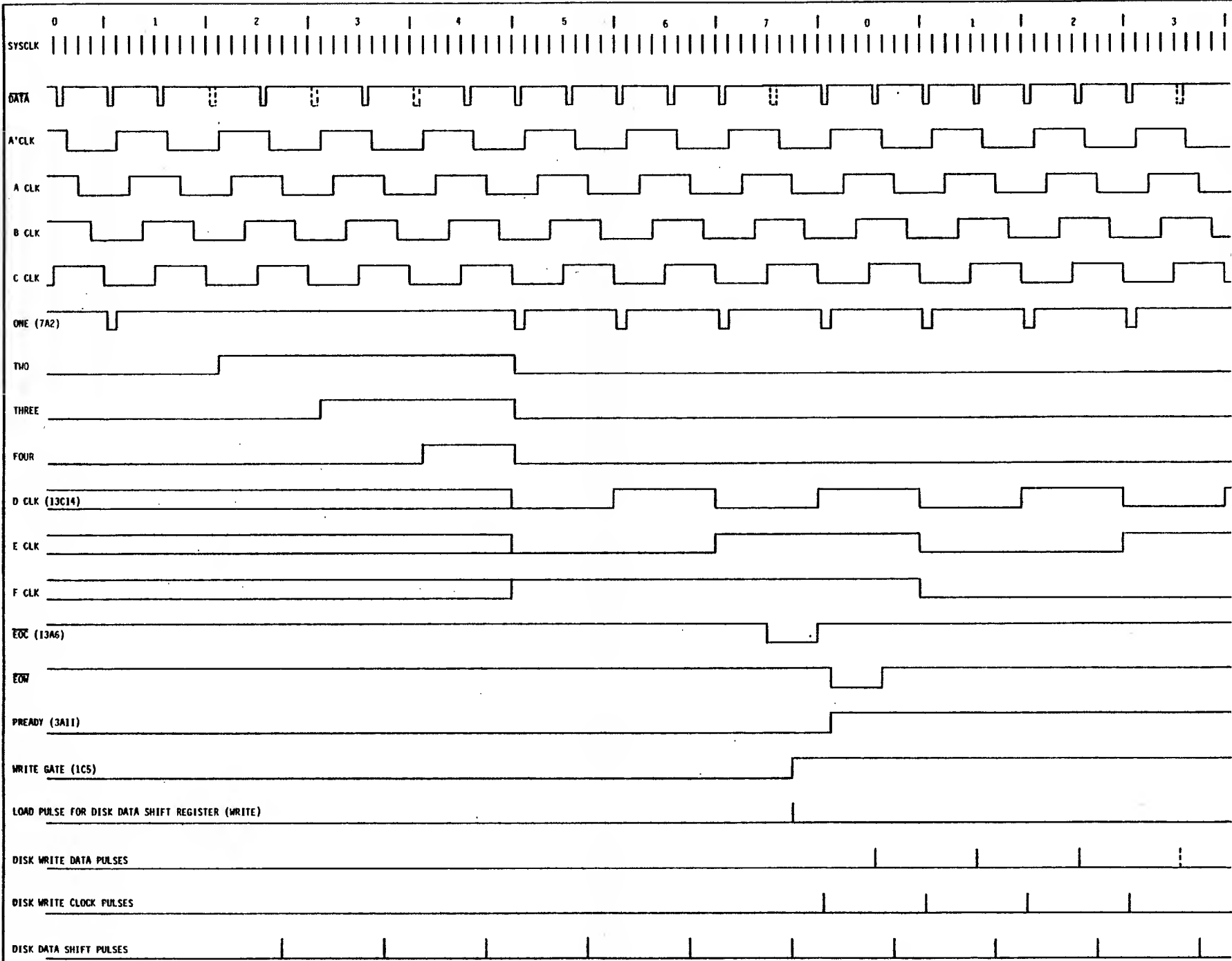
ALL RESISTANCES ARE GIVEN IN OHMS  
 ALL CAPACITANCES ARE GIVEN IN MICROFARADS



THE DISK JOCKEY I A UNIVERSAL FLOPPY DISK CONTROLLER FROM MORROW'S MICRO-STUFF  
 A THINKER TOY™ PRODUCT  
 DISK DATA BUFFERS, DISK WRITE LOGIC, AND SERIAL I/O LOGIC PAGE 3 OF 3



THE DISK JOCKEY I A UNIVERSAL FLOPPY DISK CONTROLLER FROM MORROW'S MICRO-STUFF  
 A THINKER TDY PRODUCT  
 SYSTEM BLOCK DIAGRAM



THE DISK JOCKEY IS A UNIVERSAL FLOPPY DISK CONTROLLER FROM MORROW'S MICRO-STUFF  
 A THINKER TOY PRODUCT  
 TIMING DIAGRAM FOR THE SERIAL DATA STREAM TO AND FROM THE FLOPPY DISK